Technical Report 1030

Residual Vibration Reduction in Computer Controlled Machines

Neil C. Singer

MIT Artificial Intelligence Laboratory

Residual Vibration Reduction in Computer Controlled Machines

by

Neil Cooper Singer

S.B.M.E. Massachusetts Institute of Technology
(1983)

S.M.M.E. Massachusetts Institute of Technology
(1985)

Submitted to the
Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of

Doctor Of Philosophy

at the
Massachusetts Institute Of Technology
January 1989

Signature of Author______________________________

Neil Cooper Singer
Department of Mechanical Engineering
February 10, 1989

Certified by______________________________

Professor Warren P. Seering
Committee Chairman

Accepted by______________________________

Ain A. Sonin
Chairman, Departmental Graduate Committee

Residual Vibration Reduction in Computer Controlled Machines

by

Neil Cooper Singer

Submitted to the Department of Mechanical Engineering on February 10, 1989 in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Mechanical Engineering.

Abstract

Control of machines that exhibit flexibility becomes important when designers attempt to push the state of the art with faster, lighter machines. Three steps are necessary for the control of a flexible plant. First, a good model of the plant must exist. Second, a good controller must be designed. Third, inputs to the controller must be constructed using knowledge of the system dynamic response. There is a great deal of literature pertaining to modeling and control but little dealing with the shaping of system inputs. Chapter 2 examines two input shaping techniques based on frequency domain analysis. The first involves the use of the first derivative of a gaussian exponential as a driving function template. The second, acausal filtering, involves removal of energy from the driving functions at the resonant frequencies of the system. Chapter 3 presents a linear programming technique for generating vibration-reducing driving functions for systems. Chapter 4 extends the results of the previous chapter by developing a direct solution to the new class of driving functions. A detailed analysis of the new technique is presented from five different perspectives and several extensions are presented. Chapter 5 verifies the theories of the previous two chapters with hardware experiments. Because the new technique resembles common signal filtering, chapter 6 compares the new approach to eleven standard filters. The new technique will be shown to result in less residual vibration, have better robustness to system parameter uncertainty, and require less computation than other currently used input shaping techniques.

Thesis Committee:
Prof. Warren Seering, Chairperson
Prof. John Hollerbach
Prof. Jean-Jacques Slotine
Prof. Andreas Von Flotow

Note: The methods described in Chapters 3 and 4 have been patented (pending) by the Massachusetts Institute of Technology and exclusively licensed to the author. Commercial use of these methods requires permission. The author may be contacted at: LPC, 132 Nassau St., New York, NY 10038, Phone: 212-233-2491.

Acknowledgments

Many people have helped me to complete this work. I would like to thank my friends who have generously offered their time and skills.

Prof. Warren Seering has supported me in many ways through this, and two other degrees. For this I will always be grateful.

Kenneth Pasch has been an inspiration. I only regret that he graduated and left too soon.

Professors John Hollerbach, Jean-Jacques Slotine, and Andreas von Flotow have been an excellent committee by keeping me on target during this long process.

Our research group consisting of Michael Caine, Andrew Christian, Steve Eppinger, Steven Gordon, Peter Meckl, Lukas Reuker, William Singhose, Karl Ulrich, Erik Vaaler, and Al Ward has provided an unparalleled environment for producing excellent work. The support, constructive criticism, idea input, and advice provided by all of these people were truly invaluable. I must also thank Sundar Narasimhan; his generous time commitment in designing and debugging the computer part of the hardware test rig made the experimentation possible.

My family has been a source of constant support. I would like to thank my wife, Lydia for her encouragement and faith. And, of course, my parents, Robert Singer and Dona Singer who have given me the most generous help that any parents could possibly give. I would lastly like to thank my brother, Samuel Singer, for his help throuhout the years.

The MIT Artificial Intelligence Laboratory was a wonderful, supportive place to work. I will always remember the great times that I have had there.

I also would like to thank the people at the C. S. Draper Laboratory, especially Joe Turnbull, Greg Barton, and Eli Gai.

Lastly, I would like to thank the Office of Naval Research and the American Association for Engineering Education for sponsoring me for my first four years of graduate school.

The research described in this paper was performed at the Massachusetts Institute of Technology Artificial Intelligence Laboratory. The laboratory's research is funded in part by the University Research Initiative under ONR contract N00014-86-K-0685 and in part by the Defense Advanced Research Projects Agency of the United States Department of Defense under ONR contract N00014-85-K-0124. Neil Singer has been supported by the Office of Naval Research Fellowship Program, and by C. S. Draper Laboratory's Internal Research and Development Program.

Contents

List of Figures

Introduction and Problem Description

Chapter 1

Control of machines that exhibit flexibility becomes important when designers attempt to push the state of the art with faster, lighter machines. Three steps are necessary for the control of a flexible plant. First, a good model of the plant must exist. Second, a good controller must be designed. Third, inputs to the controller must be constructed using knowledge of the system dynamic response. There is a great deal of literature pertaining to modeling and control but little dealing with the shaping of system inputs.

When a machine moves, its motion induces vibrations in its structure. At low speeds these vibrations can be ignored. At moderate-to-high speeds these vibrations become larger, and various parts of the machine no longer move the way in which they were intended to move. Many machines in industry are performance limited by these vibration problems. Some examples include robots, automated assembly and production equipment, inspection equipment, and heavy equipment.

Large machines, such as cranes, cherry-pickers, fire-ladders, and the US Space Shuttle robot arm must operate slowly in order to avoid large, dangerous vibrations. The operators of these machines must often wait between motions so that vibrations damp

out. For example, on the space shuttle, an astronaut must wait between 20 and 40 seconds for the manipulator arm to settle after a move. The space shuttle costs approximately $20,000 per minute to operate, therefore, a great deal of money can be saved by eliminating these pauses.

The space shuttle arm is also plagued by vibration induced from handling flexible payloads. Space structures, such as the long, flexible cooling panels that are to be used on the space station oscillate as they are moved into position. The handling of these payloads is extremely difficult if the vibration of the structure is not reduced.

Smaller machines are not exempt from this problem. Operators of high-speed wire-bonding machines look through a microscope while commanding the machine to connect silicon chips to package pins. These machines are idle forty percent of the time during which they are used because the commanded motion of the machine creates unacceptable levels of vibration.

The current method of dealing with structural flexibility is to avoid it. Machines are operated slowly so that flexibility is not a problem. If a higher speed is absolutely required, then the machine is designed to be more rigid. The extra structural rigidity is obtained at great expense. The design cost is high and the machine's weight is increased. The increase in weight lowers the maximum acceleration of the machine, therefore, designing around structural flexibility is a constant tradeoff.

The benefits of vibration reduction are numerous. First, wait time for vibrations to settle can be minimized. Second, systems with less damping can be successfully operated. Third, system operating speeds and cycle times can be increased. Fourth, machine life is extended because it is not exposed to harmful vibrations. Fifth, the machine's actuators would not have to "fight" the system's oscillations, therefore, less input energy is needed and smaller actuators are required. Sixth, more powerful motors could be successfully utilized on less-rigid, lower-weight machines, thus gaining a large increase in performance.

Many researchers are pursuing the goal of vibration reduction in machines. Current approaches are plagued by four major drawbacks. First, a good, often sophisticated model of the machine must be made. This is a difficult and time consuming process which can be justified for few machines. Second, special sensors must be added to the system or the system may have to be totally redesigned in order to use a particular technique. Third, a great deal of computation is required to implement the techniques. This costs both in operation time and programming and debugging time. Fourth, the vibration reduction capabilities of these techniques degrade rapidly if the system changes slightly. These four problems have restricted the use of current vibration control methods to the laboratory. The goal of this work is to develop a better approach for commanding motion of flexible computer-controlled machines.

Chapter 2 examines two input shaping techniques based on frequency domain analysis. The first involves the use of the first derivative of a gaussian exponential as an acceleration command to the system. Because the gaussian function has both a sharp frequency cutoff and is symmetric, it is chosen as a motion template. The second technique examined in this chapter is acausal filtering. A desired trajectory is generated and then notch-filtered. The filtering is performed acausally — the complete trajectory is processed prior to execution. Therefore, the unwanted frequency components can be removed without phase shift of the other frequencies.

Chapter 3 presents a linear programming technique for generating vibration-reducing driving functions for systems. The input to the system is discretized and a linear programming routine solves for the amplitude of the input function at each of the discrete times. The constraint equations for producing vibration-reducing inputs are presented. Next, additional constraints are included so that the vibration-reducing effects do not degrade under conditions of parameter uncertainty.

Chapter 4 extends the results of the previous chapter by developing a direct method for obtaining the solution to the new class of driving functions. A set of equations can be

directly solved to yield the simplest inputs that have the same vibration-reducing effects of the more complicated inputs generated in chapter 3. Next, A detailed analysis of the new technique is presented from five different perspectives including a new domain called vector diagrams. Lastly, a series of advanced applications and extensions of the basic technique are presented.

Chapter 5 verifies the theories of the previous two chapters with hardware experiments. A single degree-of-freedom system is constructed. In the first series of experiments, one flexible beam is attached to a rotating base and the theories presented are verified on this essentially one-mode system. Next joystick inputs are processed in real-time at one-kilohertz, demonstrating that the new technique requires little computation. Next, the technique is demonstrated on a two-mode system in hardware.

Because the new technique resembles common signal filtering, chapter 6 compares the new approach to eleven standard filters. Both notch and lowpass filters with finite and infinite impulse responses are considered. The common filters (all of which are designed in the frequency domain) are shown to have weak performance and require excessive computation.

1.0.1 Nonlinear System Disclaimer

Linear system theory is used to derive the new shaping technique. The applicability of this technique to nonlinear systems is not proved in this thesis. No general statement can be made regarding the application of the new technique to nonlinear systems since each nonlinearity poses unique problems. However, systems with nonlinearities that tend to appear as shifts in natural frequency seem to benefit from the new shaping technique because of the robustness to frequency uncertainty that was included in the derivation. Many simulations of different, geometrically nonlinear systems have been performed and a selection of representative data are included in support of this hypothesis. As long as the geometrically nonlinear system is varying slowly (relative to the time constant of

the system), the new shaping technique tends to work (at least on the manipulator-type systems that were considered). In addition, significant joint friction, nonlinear stiffness, and digitization are nonlinearities that are present in both the shuttle RMS simulations and the hardware experimental system. The new technique is shown to work well on both these systems. Additionally, all vibratory systems that have been considered by the author to date have been shown to benefit from the new shaping technique. While this is not a proof of the applicability of shaping to systems with extreme nonlinearities, it suggests that the new technique should not be overlooked merely because a system is nonlinear.

Frequency Domain Techniques

Chapter 2

2.1 Introduction to Frequency Domain Techniques

The first approach considered is to generate input trajectories that do have little or no energy content at the resonances of the system. Avoiding the resonances specifies that no residual vibration will be caused by the input. Naturally, vibration caused by other sources (such as disturbances) is not reduced. Energy content at the natural frequencies can be reduced in two ways. First, energy can be filtered from an input trajectory at the resonant frequencies of the system. The resulting trajectory will have a new shape, and will not excite the system resonances. Second, a new trajectory may be constructed that has no energy at the system resonances. Meckl and Seering [88] prove that residual vibration response is reduced if either of these techniques is performed. They also demonstrated that the energy removal must eliminate a range of frequencies in the neighborhood of the resonances because damping in the system will create a situation in which any of a range of frequencies will excite the system. Three approaches which have not been addressed by Meckl and Seering are considered in this chapter.

Often, a conventionally designed notch filter is proposed for input signal conditioning.

Chapter 6 will show that this approach gives poor results for several reasons. First, a causal (real time) filter distorts the phase of the resulting signal. This effect is aggravated by lengthening the filter sequence of digital filters or by increasing the order of analog or recursive filters. Therefore, efforts to improve the frequency characteristics of a filter result in increased phase distortion. Also, penalties, such as filter ringing or long move times result from attempts to improve the frequency characteristics of the notch filter.

2.2 Gaussian Shaped Input Trajectory

The first approach is to examine an open loop trajectory (or driving function) that has no energy content above the first resonance. This input function is required to have several properties. First, if it is a torque command, it must have an acceleration phase and a deceleration phase. Second, it must be able to be scaled for different moves. Third it should have a sharp frequency cutoff so that it can be used to drive the system as quickly as possible without exciting resonances.

The command shape chosen was the first derivative of a gaussian probability density function (figure 2.1):

$$f(t) = -\frac{(t-t_c)\ e^{\left(\frac{-(t-t_c)^2}{2\ \sigma^2}\right)}}{\sqrt{2}\ \pi\ \sigma^3} \quad . \tag{2.1}$$

Where σ and t_c determine the shape of the function, and t is time in seconds. This was chosen because it's shape is basically correct for driving the arm and it has a sharp frequency cutoff. Taking the magnitude of the Fourier transform of this expression yields the following function:

$$|\mathcal{F}(\omega)| = \frac{e^{\left(\frac{-\sigma^2\ \omega^2}{2}\right)}\ |\omega|}{2\ \pi} \quad , \tag{2.2}$$

in which the frequency is specified by ω. Figure 2.2 demonstrates that this function drops off quickly in frequency so it is well suited for use as a driving function. The parameter, σ, (referred to as the standard deviation in probability theory) is chosen so

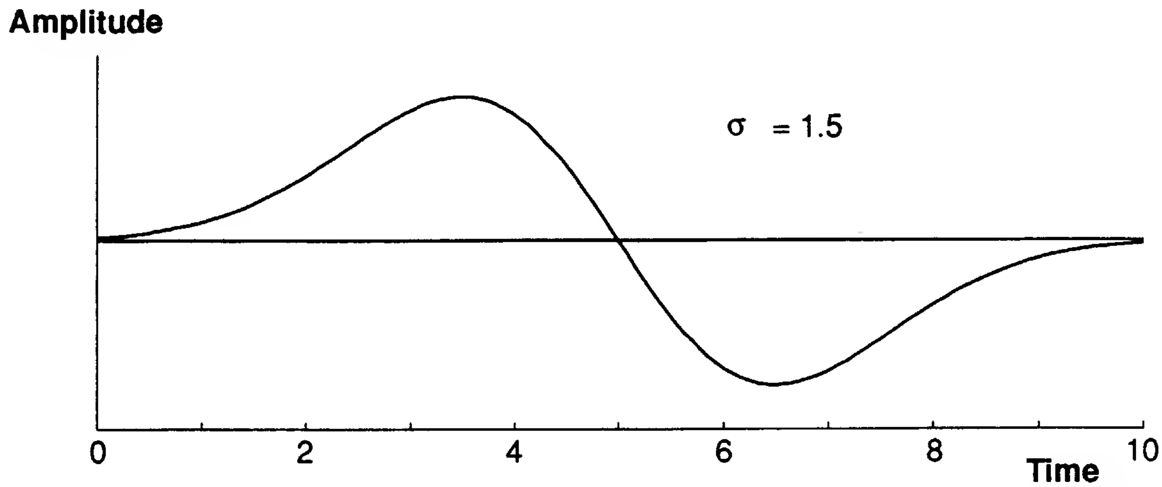


Figure 2.1: Open loop driving function. The shape is that of the first derivative of a gaussian probability density function.

that the amplitude of $\mathcal{F}(\omega)$ in equation 3.2 is small at the first resonance of the system. The time offset, t_c, is chosen to be 4σ so the driving function starts at about 2% of its maximum value. The gaussian driving function is then offset to account for friction. This is required so that the open loop torque command drives the arm only forward.

Paden and Bayo [16] have published a similar work which presents some other applications for this open loop driving function technique. Experimental results using this technique are shown in section 2.4.

2.3 Theory Verification

In order to verify the gaussian template and other methods that will be developed throughout this thesis, a test system had to be constructed. The development of good systems on which to implement various vibration reduction techniques was a significant obstacle. Several options were exercised. The first was the use of simple linear models with lumped flexibility. The second was the generation of more detailed, nonlinear models with distributed flexibility. The third was use of an existing, extremely detailed

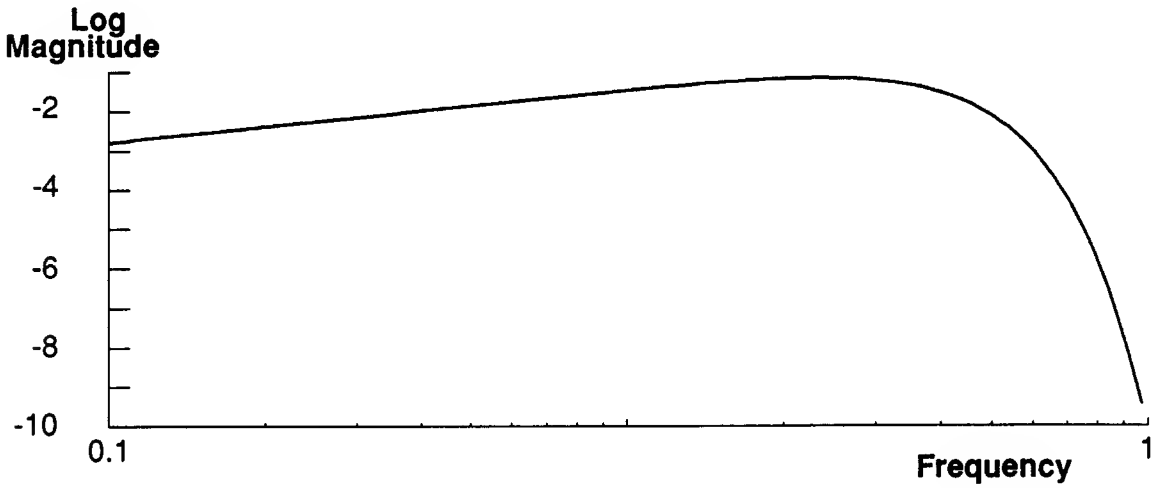


Figure 2.2: Frequency content of the gaussian driving function shown in figure 2.1.

model which includes many forms of nonlinearities and nonlinear flexibility.

2.3.1 Lumped Parameter Linear Models

Linear models are the easiest to develop and provide useful information about the potential of a new technique. Throughout this document, linear models will be used to demonstrate concepts and show the theoretical limitations of a given technique.

2.3.2 Flexible Nonlinear Models

Models that include continuous, flexible members are more realistic. However, the equations of motion for these systems are hard to obtain. For example, the assumed-mode equations of motion for a planer, two-link model with flexible links were generated by Maizza-Neto [22]. Many pages of the thesis were devoted to merely expressing the equations of motion. In addition, little insight was available from the equations because the assumed-mode terms became extremely complex.

In order to facilitate the generation of the full partial differential equation (non-assumed-mode) models for systems with continuous elements, a new extension to Kane's

method was developed. The details of this new method are given in Appendix A. Prior to this extension, Kane's method had to be used in conjunction with an assumed mode shape solution. The mode shapes of the continuous elements are assumed and ordinary differential equations are obtained. The new technique produces the exact partial differential equations of the system. The advantage of generating the exact equations are three-fold. First, more techniques are available for solving the equations. Second, the partial differential equations provide more insight into the dynamics of the system. Lastly, less mathematical manipulation is required to obtain the results.

The new method was implemented in MACSYMA, a symbolic mathematics manipulation language. Appendix B describes the code that implements the new technique. Several models were generated using the new technique.

2.3.3 The DRS Space Shuttle Manipulator Model

Next, a detailed model of the Space Shuttle Remote Manipulator System (RMS) was adapted for this research. C. S. Draper Laboratories developed this complex model which they call the DRS (Draper Remote-manipulator Simulation). NASA uses the DRS to verify and test payload operations on the actual shuttle. The Draper shuttle manipulator model includes many of the complicating features of the hardware shuttle manipulator such as stiction/friction in the joints; nonlinear gearbox stiffness; asynchronous communication timing; joint freeplay; saturation; and digitization effects. The simulation was verified with actual space-shuttle flight data. Excellent agreement was obtained both for steady-state and for transient behavior. Approximately ten man-years of programming was invested in this model in order to assure that it accurately represents the actual shuttle hardware. Detailed descriptions of the DRS model can be found in [1,54,133].

The model was executed with twenty-two degrees of freedom. These include three

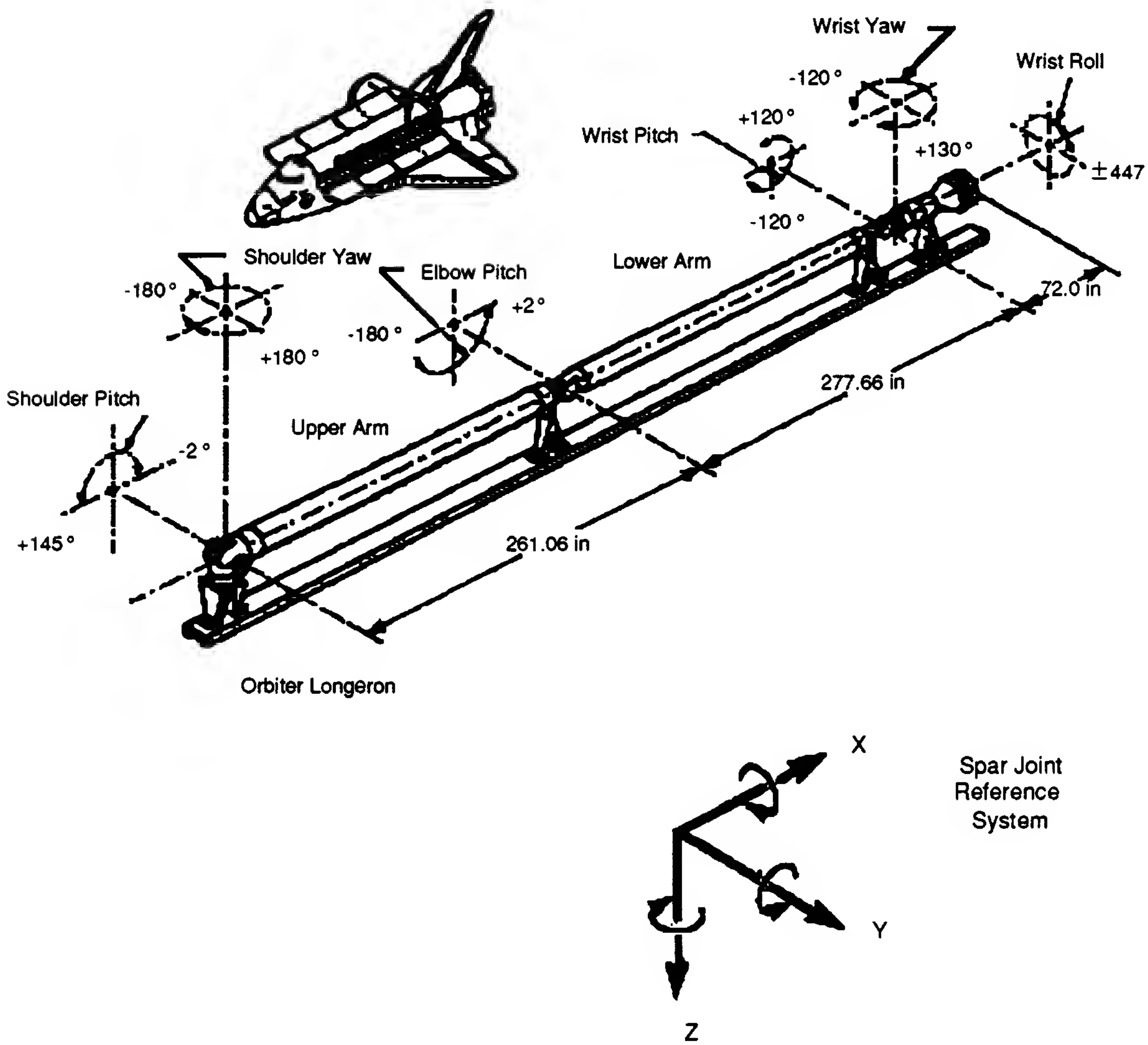


Figure 2.3: Space shuttle remote manipulator system joint reference coordinates.

rotational degrees of freedom for the space shuttle, five vibrational modes in each of the two long links, freeplay at the swingout joint and grapple point (between the arm and the payload), and seven degrees of freedom of the arm. The five vibrational modes in each long link are comprised of a first and second bending mode in two perpendicular directions, and one torsional mode. The four bending modes are modeled using an assumed cubic mode shape (figure 2.3).

The DRS model was programmed for an IBM mainframe in FORTRAN. The first task was to transfer the 14,000 lines of FORTRAN code (with 11,000 additional lines of comments) so that the program could be executed on a SUN Microsystems 3/160 computer. Various portions of the code had to altered so that it would be compatible with the UNIX FORTRAN compiler on the SUN.

Appendix C provides a detailed characterization of the DRS. This appendix provides the results of some frequency tests that were performed on the DRS model. This model was ideal as a test facility. It provided a repeatable, realistic environment for testing vibration suppression techniques. New concepts could be easily implemented in software without risking hardware. Additionally, new techniques could be inserted into the model at any location. On real hardware it is often difficult to implement different concepts because specialized hardware would have to either be altered or constructed.

2.4 Gaussian Input Trajectory Results

Figure 2.4 shows the result of driving the RMS model with both the current space shuttle controller and the open loop driving function specified in figure 2.1. The arm was moved for eight seconds in both experiments. The gaussian function drove the arm several inches further in the move time yet it left the arm with imperceptible vibrations (.08 inch maximum amplitude). The existing closed-loop controller, however, left the arm with residual vibrations over two inches in amplitude. In addition, these vibrations

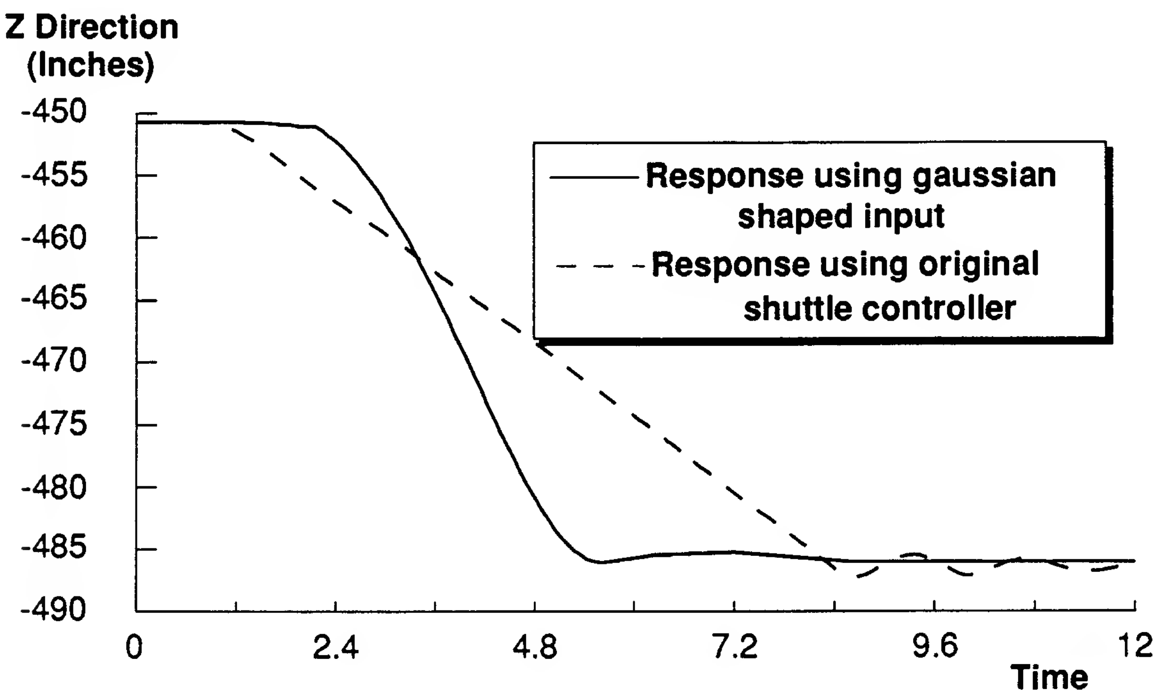


Figure 2.4: DRS response to a gaussian-shaped input compared to the response using the current shuttle controller. This is a time response of the DRS endpoint in the z-direction during a move.

remained significant ($\approx$ .3 inch amplitude) 15 seconds **after** the move ended. The residual vibration was reduced by a factor of 25 using the gaussian input.

2.5 Acausal Input Filtering

2.5.1 Shaping Inputs Through Acausal Filtering

The process of filtering a signal can be performed either "causally" or "acausally". Causal filtering refers to real-time processing of a signal — only current and past values of the signal are required for filtering. The term "acausal filtering" refers to filtering of a signal using knowledge of the entire time history — including future values. Acausal filtering can not be performed in real time, however, acausal filtering is superior to causal filtering. The advance knowledge of the entire signal that is to be filtered can

be used to correct for phase distortion and lag time. Acausal filtering can be used to process trajectories in order to remove unwanted frequency components around the system resonances. This technique may be used in any application for which trajectories are known in advance of the move.

An important consideration when discussing filtering of signals is the phase of the resulting, filtered signal. The frequency domain analysis of a signal contains information about both the magnitude and phase of the signal. While different filters will have the same effect on the magnitude, differences in the phase of the signal result in considerable differences between competing filters. The ideal is, of course, to have no phase distortion, or zero phase lag at all frequencies. Increasing phase distortion causes the filtered waveform to have less of a resemblance to the original (desired) signal.

This approach will be presented in two steps. First, a goal for an “ideal” shape will be presented. Next, a method for modifying this shape into a similar one which yields reduced vibration is discussed.

Most manipulators are velocity limited. This limitation is caused either by one or more mechanical components (ie. ball screws, linear bearings), by electronic component voltage limits, or hydraulic component flow limits. For most manipulators, this velocity limit is reached even during relatively short moves. Under these limitations the time optimal command for moving from one location to another is to accelerate as quickly as possible to this maximum velocity, coast at this speed, and then decelerate as quickly as possible to the commanded position.

First, assume that the servo controllers on the system that is to be controlled take velocity commands from the controller. Therefore, a time optimal move for the system would be commanded by a rectangular velocity pulse (figure 2.5). For other systems in which the motor torque is commanded, this rectangular pulse would be sent as a torque command to bring the system up to peak velocity as quickly as possible and then repeated in reverse to stop the moving system (figure 2.6).

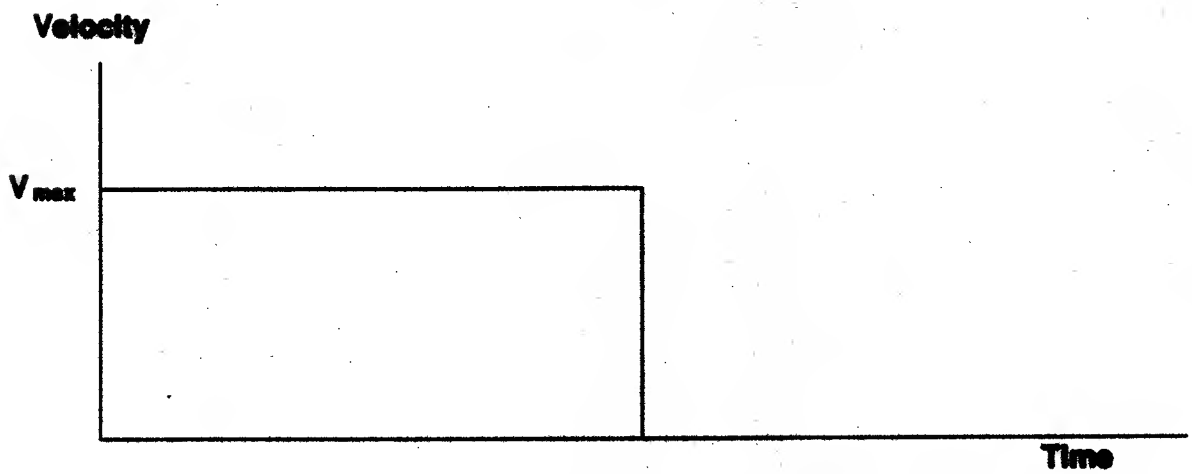


Figure 2.5: Rectangular velocity pulse given to DRS

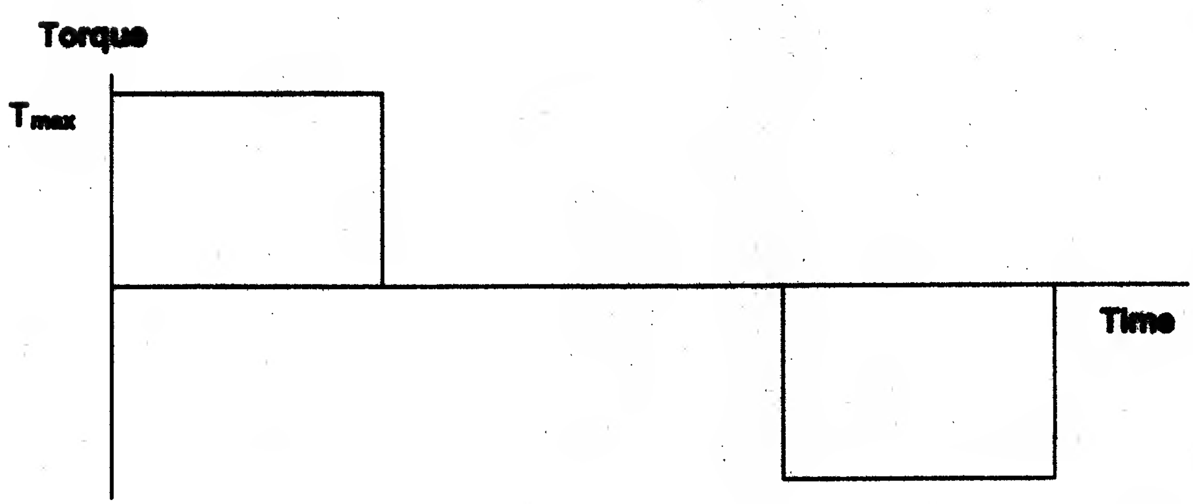


Figure 2.6: Time optimal torque command for a velocity limited system.

In either case, the rectangular torque or velocity input to the system time optimally performs the rigid body motion of the robot while simultaneously putting energy into the structural modes of the manipulator. Therefore, an acausal filter was derived for reshaping the rectangular pulse so that it contains no frequency components in the vicinity of the first two structural resonances of the manipulator.

Elimination Of Unwanted Frequencies

The elimination of energy at the system resonances is based on filter theory developed in the field of signal processing. The filters that will be designed are not intended to be used within the closed loop of the system. This strategy would not be effective since cascading a filter with the system would essentially be equivalent to adding a compensator. The root locus of the system could only be manipulated to a limited extent and the resonances would still be present. In addition, the extra significant poles added by the filter would be costly in phase lag.

The filters that will be used are for preprocessing the input to the plant so that no energy is ever put into the system near its resonances. Since in many computer-controlled machine applications there is no requirement for real-time processing, acausal filters can be utilized because of their vast superiority over causal filters. In the next section, the equation for an acausal notch filter is derived. This filter will then be used to remove the unwanted frequency components from a rectangular input pulse. The resulting shaped input pulse will have the same general shape (with added oscillations) as the rectangular pulse, however, the unwanted frequency components will have been removed.

Derivation Of The Acausal Filter

The derivation of the acausal notch filter is based on using the continuous inverse Fourier integral on a chosen analytic frequency spectrum. An approximate technique such as

using a finite impulse response (FIR) filter design program could have been used. However, an extremely fine frequency resolution would be required to successfully notch the low frequency resonances out of the space shuttle arm input. This resulted in the need for over a one thousand point FIR filter which was computationally intractable for the available computer programs [113].

First a frequency magnitude response for the filter was chosen (figure 2.7). Since the filter can be acausal, the phase lag can be chosen to be zero. The Fourier transform pair used is

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{i\omega t}d\omega \tag{2.3}$$

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt \tag{2.4}$$

where t is time, ω is frequency, $f(t)$ is the time signal and $F(\omega)$ is the Fourier transform of $f(t)$. Thus the corresponding time filter for the Fourier transform shown in figure 2.7 is given by

$$f(t) = \frac{\sin F_1 t - \sin F_2 t + \sin F_3 t - \sin F_4 + \sin F_5 t}{\pi t} \quad , \tag{2.5}$$

where F_1 and F_3 are the lower frequency edges of the the notches, F_2 and F_4 are the upper frequency edges of the two notches, and F_5 is the cutoff at the Nyquist frequency. This result can be superposed to form any combination of notches noting that a sine term corresponding to a cutoff frequency adds in the positive sense and a sine term corresponding to the start of a passband is subtracted for each additional notch.

Convolution of the Filter and a Square Pulse

The next step is to convolve the filter impulse response with the rectangular pulse of figure 2.5 given by

$$x(t) = \begin{cases} 0.0 & t < 0 \\ 1.0 & 0 < t < T_1 \\ 0.0 & T_1 < t \end{cases} \quad . \tag{2.6}$$

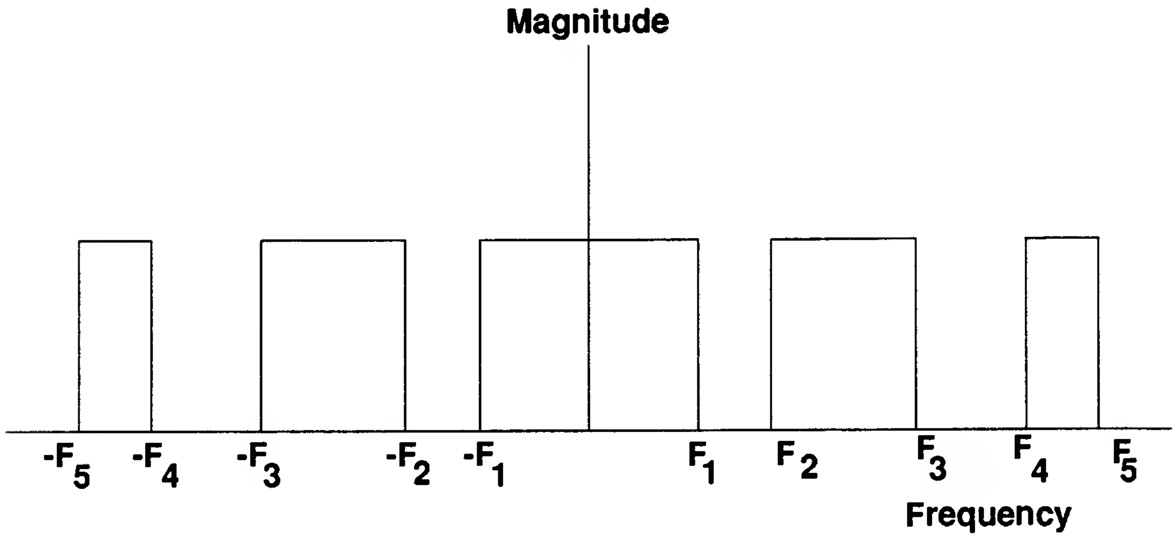


Figure 2.7: Notch filter magnitude. The phase is zero for all frequencies

The convolution expression is given by

$$y(t) = \int_{-\infty}^{\infty} f(t-\tau)x(\tau)d\tau \quad , \tag{2.7}$$

where $y(t)$ is the filtered pulse. Substituting $x(t)$ and $f(t)$ and using the relation for the sine-integral

$$\mathrm{si}(u) = -\frac{\pi}{2} + \int_0^u \frac{\sin y}{y} dy \quad , \tag{2.8}$$

equation 2.7 becomes

$$\int_0^{T_1} f(t-\tau)d\tau = \frac{g(t) - g(t-T_1)}{\pi} \quad . \tag{2.9}$$

with

$$g(x) = \mathrm{si}(F_1 x) - \mathrm{si}(F_2 x) + \mathrm{si}(F_3 x) - \mathrm{si}(F_4 x) + \mathrm{si}(F_5 x) \quad .$$

Figure 2.8 shows the result achieved by convolving the notch of figure 2.7 with a rectangular pulse. The filtered waveform exists for all time. Therefore, implementation requires that the waveform be windowed in time so that a causal robot input is generated. For the following experiments, a Bingham style cosine taper data window [101] was

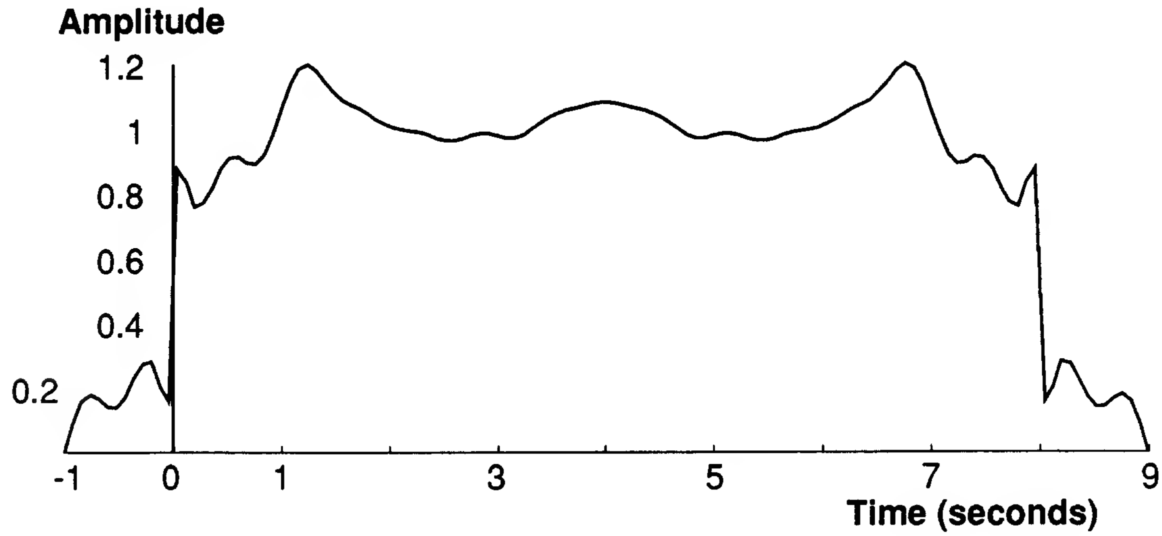


Figure 2.8: Rectangular pulse notch filtered between .4 and .8 hertz and between 1.2 and 1.7 hertz.

chosen. The cosine taper window provides a quick decay at the tails of the window, yet it does not significantly alter the spectrum of the signal. The waveform was also shifted slightly in time so that it starts at zero time (figure 2.9).

A first point to note is that if a specified filter spectrum has a very sharp transition, the impulse (time) response becomes long. This is accommodated by windowing after the filter is convolved with another waveform. A gentle transition in the frequency spectrum (possibly caused by windowing in the frequency domain) would result in a more narrow impulse response and, in some cases, would eliminate the need for windowing later. A sharp notch was used for simplicity since the benefits of windowing in frequency, rather than time, were not clear in this application.

A second point to note is that this derivation was performed analytically, rather than using a Discrete Fourier Transform (DFT). This avoided any effects of discretization and truncation until the end. The "ideal" filter could first be generated and than processed into a discrete waveform for input into the system.

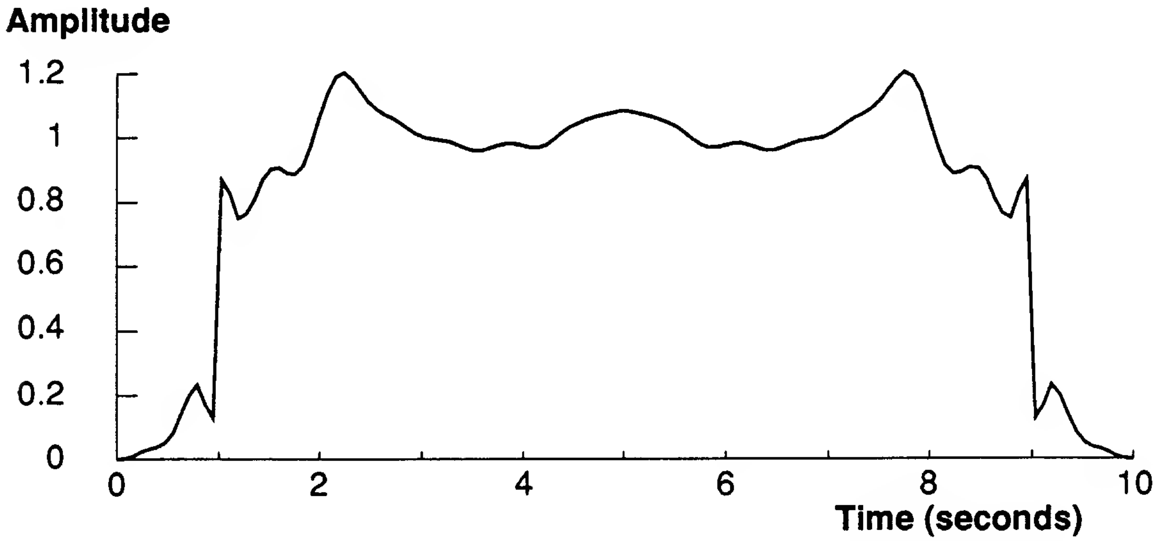


Figure 2.9: Windowed and shifted rectangular pulse notch filtered between .4 and .8 hertz; and between 1.2 and 1.7 hertz.

Computational Issues

The calculation of these filtered rectangular pulses requires the computation of the sine integral. This computation poses some numerical difficulties. The series approximation of $\mathrm{si}(u)$ is an alternating series. For moderate and large values of u (u greater than 15), the terms become huge and precision is lost.

Romberg integration of equation 2.8 also proved to fail because the integrand is periodic and the Romberg algorithm uses constant interval sizes, thus accumulating errors. The random interval romberg integration implemented on Hewlett-Packard™ calculators was the only standard routine found that was successful in maintaining precision integrating over many periods of the function.

The waveforms were calculated by numerically integrating $\mathrm{si}(u)$ once over each half period (intervals of π) with a standard Romberg technique. These values were stored in a table. The value of si was looked up at the multiple of π nearest to u. The final value of $\mathrm{si}(u)$ was computed by adding to this the integral from the multiple of π to u.

This technique has the advantage that most of the numerical integration is done once and stored. In addition, the Romberg integration is never used to integrate over more than one half of one period of the integrand, so there is no loss of precision.

Once the analytical input function is generated, a wide range of signal processing issues must be addressed prior to implementation. This involves careful consideration of several tradeoffs. By truncating the resulting infinite continuous function for the input, the carefully chosen frequency characteristics are altered. If the function is truncated to be short in time (in order to reduce move time), the frequency notches tend to close. This is because the time function is being multiplied by a short duration window, which is equivalent to convolution with a long duration signal in the frequency domain. A smooth time domain windowing function is more desirable than a sharp one. This is because a sharp cutoff causes large, infinite duration ripples in the frequency spectrum of the input waveform. However, a smooth window tends to make the time duration longer. Lastly, the actual Fourier transform of the discrete signal increases between the frequency points. In order to reduce this effect, the solution is to close the spacing of the points in the frequency domain. However, improving the resolution of the frequency response in the notches results in an increased time duration. In summary, improvements in time domain response conflicts with improvements in the frequency domain. This conclusion is well documented in the field of digital signal processing [101,128].

Acausal Filtering Summary

The use of shaped inputs for controlling open or closed loop plants shows that significant vibration reduction can be achieved. The cost in extended move time is small compared to the time saved in settling of the robot. Since settling time often limits the completion of a useful task, this is an important problem to address.

By using acausally filtered rectangular pulses to control a plant, a close approximation to the time-optimal, rectangular waveform can be generated. This new waveform

can then be used to drive a manipulator without significantly exciting its structural resonances.

This technique has some limitations due to the process by which the input waveforms are generated. There is a constant tradeoff between the time length of the shaped input and the quality of its vibration reducing properties. Therefore, signal processing issues such as windowing are important to consider when using this technique.

2.6 Acausal Filter Results

Figure 2.10 shows a comparison between the current shuttle RMS controller and an acausally filtered input controller. A step command in velocity is given to the manipulator at one joint. The RMS controller produces a ramped input to the motor. The acausally filtered controller generates the motor command shown in figure 2.9. Shown is the motion of the endpoint of the manipulator throughout the move. The residual vibration is reduced by a factor of four for the unloaded shuttle arm.

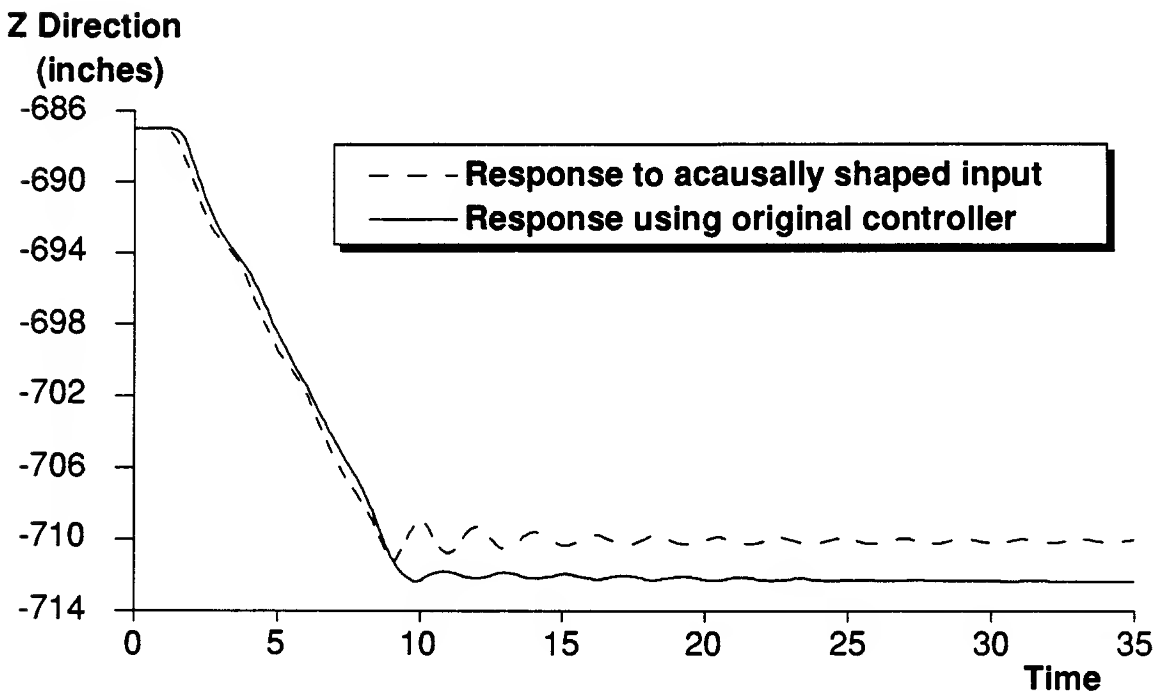


Figure 2.10: Comparison between the RMS controller and an acausally filtered rectangular input. The two runs do not end at the exact same location because both moves are open-loop command responses.

Generating a Simple, Vibration-Free Input Function

Chapter 3

3.1 Introduction

Most researchers have examined the transient vibration of manipulators in terms of frequency content of the system inputs and outputs. This approach inherently assumes that the system inputs are not actually transient, but are one cycle of a repeating waveform. The approach taken in this chapter is fourfold: first, the transient residual vibration amplitude of a system will be directly expressed as a function of its transient input. Second, the input will be specified so that the system's natural tendency to vibrate is used to cancel residual vibration. Third, the input will be modified to include robustness to uncertainties. Fourth, the case of arbitrary system inputs will be examined.

The analyses and derivations presented in this Chapter are based on assumptions of system linearity. In the next chapter and appendix D some of the linearity assumptions will be relaxed and this technique will be shown to work on some non-linear systems. The robustness that is included in the linear derivation enables this technique to work on many systems for which superposition does not hold (including real harware).

This chapter considers the problem of generating some input command that completes a motion of the system without any residual vibration. Several researchers have produced vibration-reducing input functions (see Chapters 2 and 7; also [45,130,49,63]). These approaches involved selecting parameters in continuous template functions. This chapter approaches the problem from a new perspective. A discrete input is considered and the values of the input function at each time is determined.

The motivation for this approach was the use of the DRS shuttle manipulator model described in section 2.3. Command updates are sent to the servos every .08 seconds. At this rate, an input trajectory contains relatively few points (only 12.5 points per second). The approach that was used was to assign a variable to each of these servo commands and to solve for a set of these commands that moves the system without any residual vibration. The duration of the input was shortened (the number of points or variables was reduced) until the shortest vibration-free input was obtained. Figure 3.1 demonstrates this concept graphically. The advantage of this approach is that input functions do not need to be continuous. This chapter details a series of variations on this technique, several of which were used to command the DRS model.

3.2 Linear Programming Formulation

The method of generating a digital input sequence is to rely on either an integer or linear programming routine to select the input values (the A_j in figure 3.1). For the space shuttle manipulator, sequences were solved using the following linear programming formulation. A variable, A_j is established for the value of each input pulse. For example, if a 2 second sequence is to be generated with .1 second sampling times then 20 variables will be selected. The next step is to formulate an expression that relates the A_j to the residual vibration of a system.

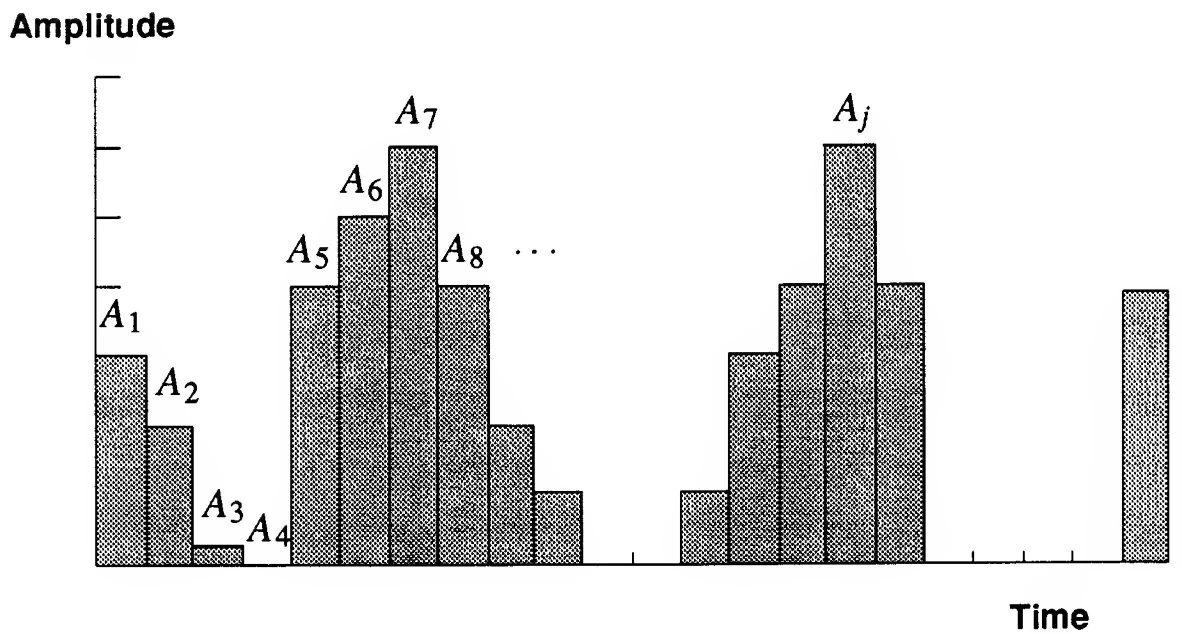


Figure 3.1: Concept of discrete input sequences. The unknown amplitudes, A_j, in the impulse sequence shown are computed by the optimization routines described in this chapter. The width of the pulses may be set by the digitization of the system that is to be controlled or some other system limitation.

3.2.1 No Vibration Constraint Equations

The first step toward generating constraint equations which guarantee a vibration-free system output is to specify the system response to an impulse input. For this derivation, the system input is assumed to be a position command and the output, an actual position. The results of this section generalize for any input or output (velocity, torque, etc.). A linear, vibratory system of any order can be specified as a cascaded set of second-order poles with the decaying sinusoidal response [19]:

$$y(t) = \left[A \frac{\omega}{\sqrt{1.0 - \zeta^2}} e^{-\zeta\omega(t-t_0)} \right] \sin \omega\sqrt{1.0 - \zeta^2}(t - t_0) \quad , \tag{3.1}$$

where A is the amplitude of the position impulse command, ω is the undamped natural frequency of the plant, ζ is the damping ratio of the plant, t is time, and t_0 is the time of the impulse input. Equation 3.1 specifies the position response, $y(t)$, at some point of interest in the system. This equation also assumes that numerator dynamics are not present. The validity of this assumption is discussed in section 4.5.5. In this section, only one mode is assumed (the general case is treated in section 4.6).

Equation 3.1 is the system response to an impulse input. The response of the system to a pulse is a trivial extension of this since the digital pulse input can be formed from a superposition of impulses. The response to the pulse is, therefore, the superposition of responses given by equation 3.1. For the purposes of the derivation in this section, the finite duration pulse can be treated as if it were the equivalent impulse which has the same area and occurs at the starting time of the pulse. Once the result is obtained, the validity of this simplification will be discussed (section 4.4). In this section, only one mode is assumed (the general case is treated in section 4.6). The two impulse responses (each described by (3.1)) can be expressed mathematically for all times greater than the duration of the input. The two decaying sinusoidal responses can be added using the trigonometric relation:

$$B_1 \sin(\alpha t + \phi_1) + B_2 \sin(\alpha t + \phi_2) = A_{\text{amp}} \sin(\alpha t + \psi) \quad , \tag{3.2}$$

where

$$\begin{aligned} A_{\text{amp}} &= \sqrt{(B_1 \cos\phi_1 + B_2 \cos\phi_2)^2 + (B_1 \sin\phi_1 + B_2 \sin\phi_2)^2} \\ \psi &= \tan^{-1}\left(\frac{B_1 \cos\phi_1 + B_2 \cos\phi_2}{B_1 \sin\phi_1 + B_2 \sin\phi_2}\right) \quad , \end{aligned}$$

This equation can be generalized to the N impulse case by applying the formula recursively and simplifying the result (or more simply, by noting the symmetry of the problem and proving by induction). The amplitude of vibration for a multi-impulse input is given by (from [48]):

$$\begin{aligned} A_{\text{amp}} &= \sqrt{\left(\sum_{j=1}^{N} B_j \cos\phi_j\right)^2 + \left(\sum_{j=1}^{N} B_j \sin\phi_j\right)^2} \\ \phi_j &= \omega\sqrt{1-\zeta^2}\, t_j \quad . \end{aligned} \tag{3.3}$$

The B_j are the coefficients of the sine term in (3.1) for each of the N impulse inputs, and the t_j are the times at which the impulses occur. Elimination of vibration after the input has ended requires that the expression for A_{amp} equal zero at the time at which the input ends, t_{end}. This is true if both squared terms in 3.3 are independently zero, yielding:

$$B_1 \cos\phi_1 + B_2 \cos\phi_2 + \cdots + B_N \cos\phi_N = 0 \tag{3.4}$$

$$B_1 \sin\phi_1 + B_2 \sin\phi_2 + \cdots + B_N \sin\phi_N = 0 \quad , \tag{3.5}$$

with

$$B_j = \frac{A_j \omega}{\sqrt{1-\zeta^2}} e^{-\zeta\omega(t_{end}-t_j)} \quad ,$$

where A_j is the amplitude of the jth impulse, t_j is the time of the jth impulse, and t_{end} is the time at which the sequence ends (time of the last impulse). Making the final

substitution and simplifying yields:

$$
\begin{aligned}
V_1 &= \sum_{j=1}^{N} A_j e^{-\zeta\omega(t_{end}-t_j)} \sin\left(t_j\omega\sqrt{1-\zeta^2}\right) = 0 \\
V_2 &= \sum_{j=1}^{N} A_j e^{-\zeta\omega(t_{end}-t_j)} \cos\left(t_j\omega\sqrt{1-\zeta^2}\right) = 0
\end{aligned}
\tag{3.6}
$$

If the input is chosen so that there are N digital pulses, N terms must be included in equation 3.6.

These two equations can now be used to generate an input sequence for a digital system with some known resonance. A revised simplex algorithm (IMSL Routine ZX3LP) was used to generate sequences of N digital impulses. The formulation used was to minimize the sum of V_1 and V_2 given by the equations 3.6. The number of pulses in the input was shortened until V_1 and V_2 were no longer zero. The result was the shortest sequence that met this no-vibration constraint.

The constraints given by equations 3.6 guarantee no residual vibration as long as the assumption of the system natural frequency and damping ratio are correct. Since the parameters of the system are never precisely known, some robustness to system uncertainty must be incorporated. Two additional constraint equations are added to provide this robustness to the solutions that are obtained. Further discussion and justification of these constraints are given in chapter 4. The robustness constraint that was chosen can be expressed in English as "have the residual vibration change little for variations in the natural frequency or damping ratio of the system". Mathematically this constraint is formed by taking the first derivative of equation 3.3 with respect to ω and ζ (the system natural frequency and damping ratio). However, it can be easily shown that this is mathematically equivalent to taking the derivative of equations 3.6 with respect to ω. The next section proves that setting the derivative with respect to ω to zero automatically sets the derivative with respect to ζ to zero.

3.2.2 The Robustness Constraint Equations

The derivatives of the expressions for V_1 and V_2 (equation 3.6) are taken with respect to ω. This step results in the following equations:

$$\begin{aligned}\frac{dV_1}{d\omega} &= \sum_{j=1}^{N} -A_j\omega t_j e^{-\zeta\omega(t_{end}-t_j)} \sin\left(t_j\omega\sqrt{1-\zeta^2}\right)\\ &\quad - \frac{A_j\zeta\omega\left(t_{end}-t_j\right)e^{-\zeta\omega(t_{end}-t_j)}\cos\left(t_j\omega\sqrt{1-\zeta^2}\right)}{\sqrt{1-\zeta^2}}\\ &\quad + \frac{A_j e^{-\zeta\omega(t_{end}-t_j)}\cos\left(t_j\omega\sqrt{1-\zeta^2}\right)}{\sqrt{1-\zeta^2}}\end{aligned} \tag{3.7}$$

$$\begin{aligned}\frac{dV_2}{d\omega} &= \sum_{j=1}^{N} A_j\omega t_j e^{-\zeta\omega(t_{end}-t_j)} \cos\left(t_j\omega\sqrt{1-\zeta^2}\right)\\ &\quad - \frac{A_j\zeta\omega\left(t_{end}-t_j\right)e^{-\zeta\omega(t_{end}-t_j)}\sin\left(t_j\omega\sqrt{1-\zeta^2}\right)}{\sqrt{1-\zeta^2}}\\ &\quad + \frac{A_j e^{-\zeta\omega(t_{end}-t_j)}\sin\left(t_j\omega\sqrt{1-\zeta^2}\right)}{\sqrt{1-\zeta^2}}\end{aligned} \tag{3.8}$$

These equations can be simplified. First, the last term of each of equations 3.7 and 3.8 can be multiplied by ω to yield equations 3.4 and 3.5 which are the equations for no residual vibration. These terms must be zero. (The prerequisite for this robustness process was that there be no residual vibration.)

The second term of each of equations 3.7 and 3.8 results from the change of the damping envelope with changes in ω. This term (as will be shown below) surprisingly also becomes zero. The result after dividing both sides by ω is

$$\begin{aligned}\sum_{j=1}^{N} A_j t_j e^{-\zeta\omega(t_{end}-t_j)} \sin\left(t_j\omega\sqrt{1-\zeta^2}\right) &= 0\\ \sum_{j=1}^{N} A_j t_j e^{-\zeta\omega(t_{end}-t_j)} \cos\left(t_j\omega\sqrt{1-\zeta^2}\right) &= 0 \quad .\end{aligned} \tag{3.9}$$

These two equations together form the first-derivative robustness criterion.

The proof of this answer is not straightforward because the results from the sinusoidal terms are used to simplify the cosinusoidal terms and vice-versa. This fact requires that we assume an answer and prove that (if the answer is correct) the results are consistent.

Now, using this assumed result from equation 3.9, the second term of equation 3.7 will be shown to be exactly zero.

$$\begin{aligned} \text{Term} &= -\frac{A_j\zeta\omega\left(t_{end}-t_j\right)e^{-\zeta\omega(t_{end}-t_j)}\cos\left(t_j\omega\sqrt{1-\zeta^2}\right)}{\sqrt{1-\zeta^2}} \\ &= -\frac{A_j\zeta\omega t_{end}e^{-\zeta\omega(t_{end}-t_j)}\cos\left(t_j\omega\sqrt{1-\zeta^2}\right)}{\sqrt{1-\zeta^2}} \\ &\quad +\frac{A_j\zeta\omega t_j e^{-\zeta\omega(t_{end}-t_j)}\cos\left(t_j\omega\sqrt{1-\zeta^2}\right)}{\sqrt{1-\zeta^2}} \end{aligned} \tag{3.10}$$

By requiring that $V_1 = 0$, the first term of equation 3.10 must be zero (from equation 3.6). The term that remains is exactly the second expression from equation 3.9 (the term that was assumed to be zero!)

The same dependent proof is required to show that the derivative with respect to ζ is already satisfied by satisfying equations 3.9 and 3.6. Start by taking the derivatives of equations 3.6 with respect to ζ:

$$\begin{aligned} \frac{dV_1}{d\zeta} &= \sum_{j=1}^{N}\frac{A_j\zeta\omega^2 t_j e^{-\zeta\omega(t_{end}-t_j)}\sin\left(t_j\omega\sqrt{1-\zeta^2}\right)}{1-\zeta^2} \\ &\quad -\frac{A_j\omega^2\left(t_{end}-t_j\right)e^{-\zeta\omega(t_{end}-t_j)}\cos\left(t_j\omega\sqrt{1-\zeta^2}\right)}{\sqrt{1-\zeta^2}} \\ &\quad +\frac{A_j\zeta\omega e^{-\zeta\omega(t_{end}-t_j)}\cos\left(t_j\omega\sqrt{1-\zeta^2}\right)}{\left(\sqrt{1-\zeta^2}\right)^3} \end{aligned} \tag{3.11}$$

$$\frac{dV_2}{d\zeta} = \sum_{j=1}^{N}\frac{-A_j\zeta\omega^2 t_j e^{-\zeta\omega(t_{end}-t_j)}\cos\left(t_j\omega\sqrt{1-\zeta^2}\right)}{1-\zeta^2}$$

$$
-\frac{A_j\omega^2\left(t_{end}-t_j\right)e^{-\zeta\omega(t_{end}-t_j)}\sin\left(t_j\omega\sqrt{1-\zeta^2}\right)}{\sqrt{1-\zeta^2}}
+\frac{A_j\zeta\omega e^{-\zeta\omega(t_{end}-t_j)}\sin\left(t_j\omega\sqrt{1-\zeta^2}\right)}{\left(\sqrt{1-\zeta^2}\right)^3} \tag{3.12}
$$

Each of the terms in these two equations can be eliminated by using the identities of equations 3.9 and 3.6. This proves that setting both the residual vibration constraints to zero and the constraints that set the derivatives of these constraints with respect to ω to zero causes the derivative expressions with respect to ζ to be zero.

Higher derivative constraints follow by differentiating equations 3.6 further. The general result is that the qth derivative constraint is given by:

$$
\begin{aligned}
\sum_{j=1}^{N} A_j(t_j)^q e^{-\zeta\omega(t_{end}-t_j)}\sin\left(t_j\omega\sqrt{1-\zeta^2}\right) &= 0\\
\sum_{j=1}^{N} A_j(t_j)^q e^{-\zeta\omega(t_{end}-t_j)}\cos\left(t_j\omega\sqrt{1-\zeta^2}\right) &= 0
\end{aligned} \tag{3.13}
$$

The next section describes how these equations are included in an optimization routine in order to generate a robust, vibration-reducing input sequence.

3.2.3 Assembling the Linear Programming Constraints

The constraints given above can now be used to form a robust, vibration-less system command input. The same revised simplex program that was used in the previous section was used to solve for the new sequences. The second derivative expression (equation 3.13 with $q = 2$) given by:

$$
\begin{aligned}
\sum_{j=1}^{N} A_j(t_j)^2 e^{-\zeta\omega(t_{end}-t_j)}\sin\left(t_j\omega\sqrt{1-\zeta^2}\right) &= 0\\
\sum_{j=1}^{N} A_j(t_j)^2 e^{-\zeta\omega(t_{end}-t_j)}\cos\left(t_j\omega\sqrt{1-\zeta^2}\right) &= 0
\end{aligned} \tag{3.14}
$$

is then minimized subject to the following constraints:

- The sinusoidal part of the vibration amplitude expression equals zero (the first equation of 3.6).
- The cosinusoidal part of the vibration amplitude expression equals zero (the second equation of 3.6).
- The sinusoidal part of the first derivative ($\frac{d}{d\omega}$) expression equals zero (the first equation of 3.9).
- The cosinusoidal part of the first derivative ($\frac{d}{d\omega}$) expression equals zero (the second equation of 3.9).
- The magnitude of the impulse amplitudes must be less than a limit ($A_j <=$ Limit)
- The sum of the impulse amplitudes are unity.

The length of the input sequence is determined by the number of variables that are chosen (which is equal to the number of terms in the summations). The length of sequence (N in the equations above) is reduced until the constraint equations can no longer be satisfied. The resulting solutions are theoretically exact. If the system were to be exactly as modeled, the response to the input would be totally without vibration. The digital timing of the system is already included in the derivation, therefore, the digitization does not alter the vibration-reducing effects. For example, the space shuttle manipulator in a particular configuration has a .5 hz natural frequency. The digital timing of the controller is fixed at .08 seconds. The smallest sequence that meets the constraints above is 22 digital pulses in length (1.68 seconds). The resulting sequence is shown in figure 3.2.

The unusual result that was obtained is that the input sequence does not approach a continuous function, but rather is a sequence of discrete pulses. The understanding of these results (the topic of chapter 4) leads to a better technique for commanding systems

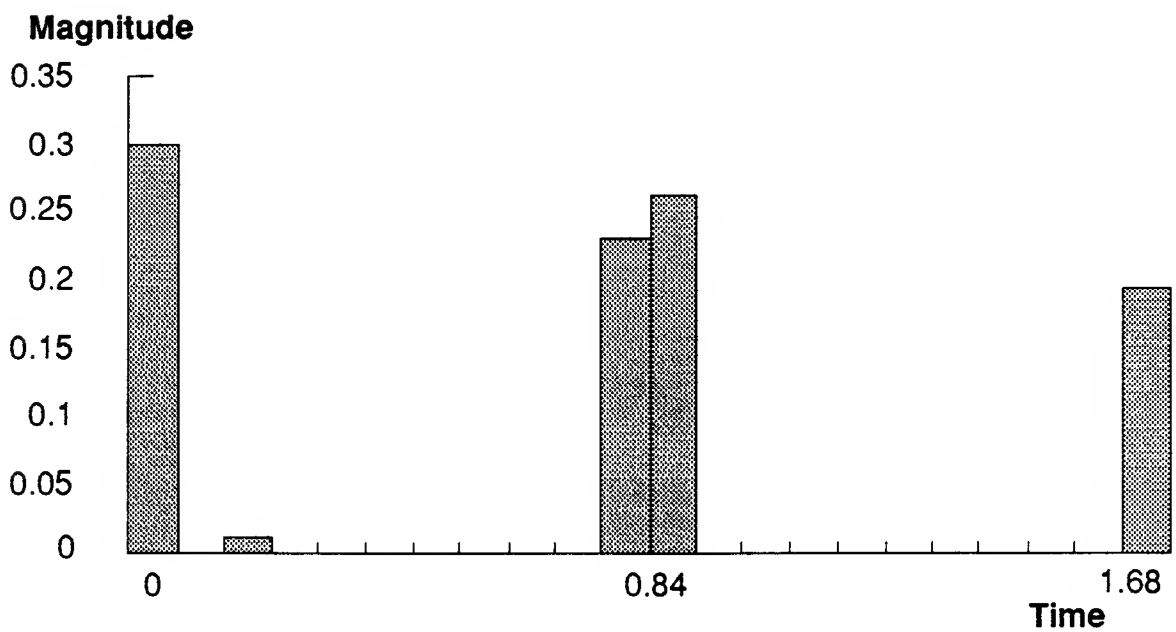


Figure 3.2: Robust digital sequence. This sequence meets the constraint that requires that the system have no residual vibration when the input has ended. Additionally, this sequence meets the robustness constraint that requires the rate of change of the vibration with respect to changes in natural frequency be zero. Therefore, small uncertainties in the parameters of the system (ie. natural frequency) do not cause an appreciable increase in residual vibration.

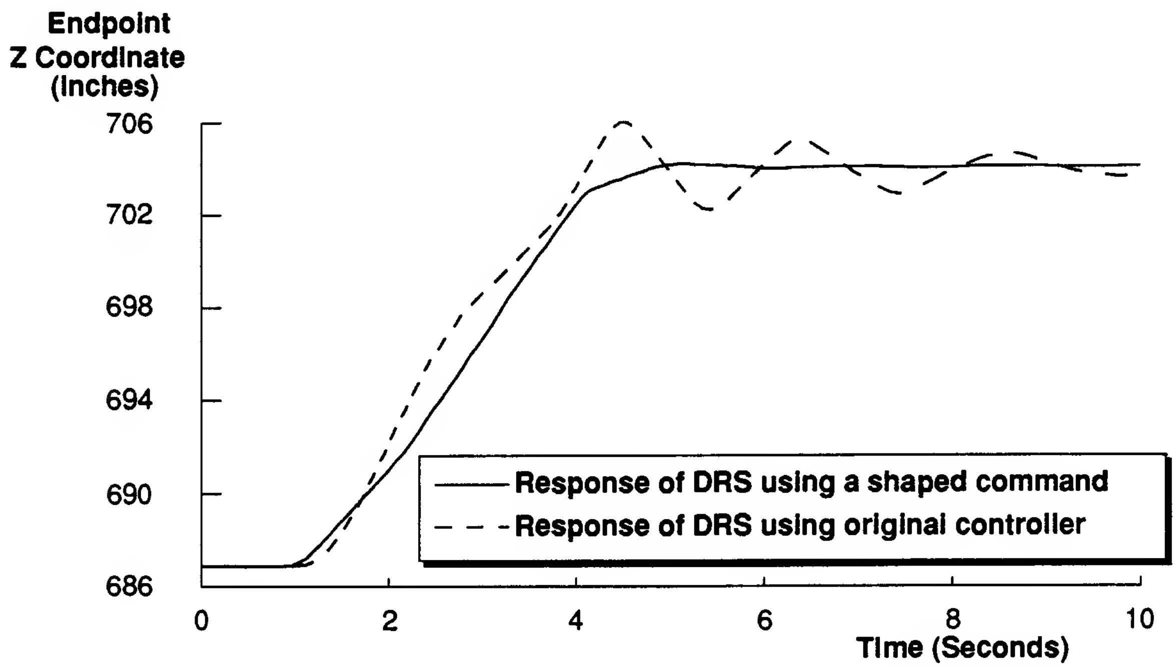


Figure 3.3: Comparison between the RMS controller and a controller that shapes inputs with a three-impulse equivalent sequence (a second-order robust sequence adjusted for the digital system).

to move without vibration. In addition, a method for using a basic, vibration-reducing input to form arbitrary vibration-reducing inputs for a system will also be presented in the next chapter.

3.3 Results on the DRS Model

The shaped command of the previous section was next tested on the computer model of the Space Shuttle Remote Manipulator System (RMS). The details of this and the other models used for theory verification are given in section 2.3. Figure 3.3 shows a comparison between the response of the DRS using the current shuttle RMS controller and the response of the DRS using the digital sequence of figure 3.2 as a velocity input. The residual vibration is reduced by more than one order of magnitude for the unloaded shuttle arm. Comparable results were obtained for a variety of moves tested.

3.4 Using Additional Constraints

This method of generating input sequences can be implemented using more constraints than the ones chosen in the analysis above. (The constraints of zero vibration and zero derivatives to changes in natural frequency.) The advantage of choosing other constraints is that the input can be tailored to a specific system or application. As an example a sequence with much greater robustness will be generated in this section. The additional robustness could be obtained by setting higher derivatives to zero, however, (as will be shown in chapter 4) setting higher derivatives to zero results in longer duration inputs. An alternate route is to enforce constraints that minimize vibration over a range of frequencies. The resulting vibration is not necessarily zero but is forced to be small.

A variety of different constraint combinations can be used. An example is given for a constraint problem that allows the amplitudes of the pulses to be negative. This sequence is created with the following constraints:

- The sinusoidal part of the vibration amplitude expression equals zero at the frequency ω_n (the first equation of 3.6).
- The cosinusoidal part of the vibration amplitude expression equals zero at the frequency ω_n (the second equation of 3.6).
- The sinusoidal part of the first derivative (w.r.t natural frequency) expression at the frequency ω_n equals zero (the first equation of 3.9).
- The cosinusoidal part of the first derivative (w.r.t natural frequency) expression at the frequency ω_n equals zero (the second equation of 3.9).
- The sinusoidal part of the vibration amplitude expression is less than some threshold value at the frequency ω_1 (the first equation of 3.6).

- The cosinusoidal part of the vibration amplitude expression is less than some threshold value at the frequency ω_1 (the second equation of 3.6).

- The sinusoidal part of the vibration amplitude expression is less than some threshold value at the frequency ω_2 (the first equation of 3.6).

- The cosinusoidal part of the vibration amplitude expression is less than some threshold value at the frequency ω_2 (the second equation of 3.6).

- The magnitude of the impulse amplitudes must be less than a limit ($|A_j| <=$ Limit)

- The sum of the impulse amplitudes are unity.

An example of utilizing the constraints given above is shown in figure 3.4. The vibration error curve shown was generated by constraining the vibration error and slope at three nearby frequencies. The effect of this is to create a sequence that suppresses residual vibration over a large range of frequencies. The price that is paid for the additional constraints is often duration of the sequence in time. However, it is important to note that the sequence with the vibration error plot shown in figure 3.4 is shorter in duration than the sequence obtained by setting several derivatives to zero in order to achieve the same robustness. This is because the vibration error is not constrained to be zero throughout the region of interest — a minimum level of tolerable vibration was set and the sequence was constrained to remain below that level. Depending on the application, this technique may be preferable to the simpler derivative sequences of section 4.3.

3.5 Summary

Optimization routines that used constraints on the time-domain behavior of a sample system were used to generate input sequences for that system. The resulting inputs

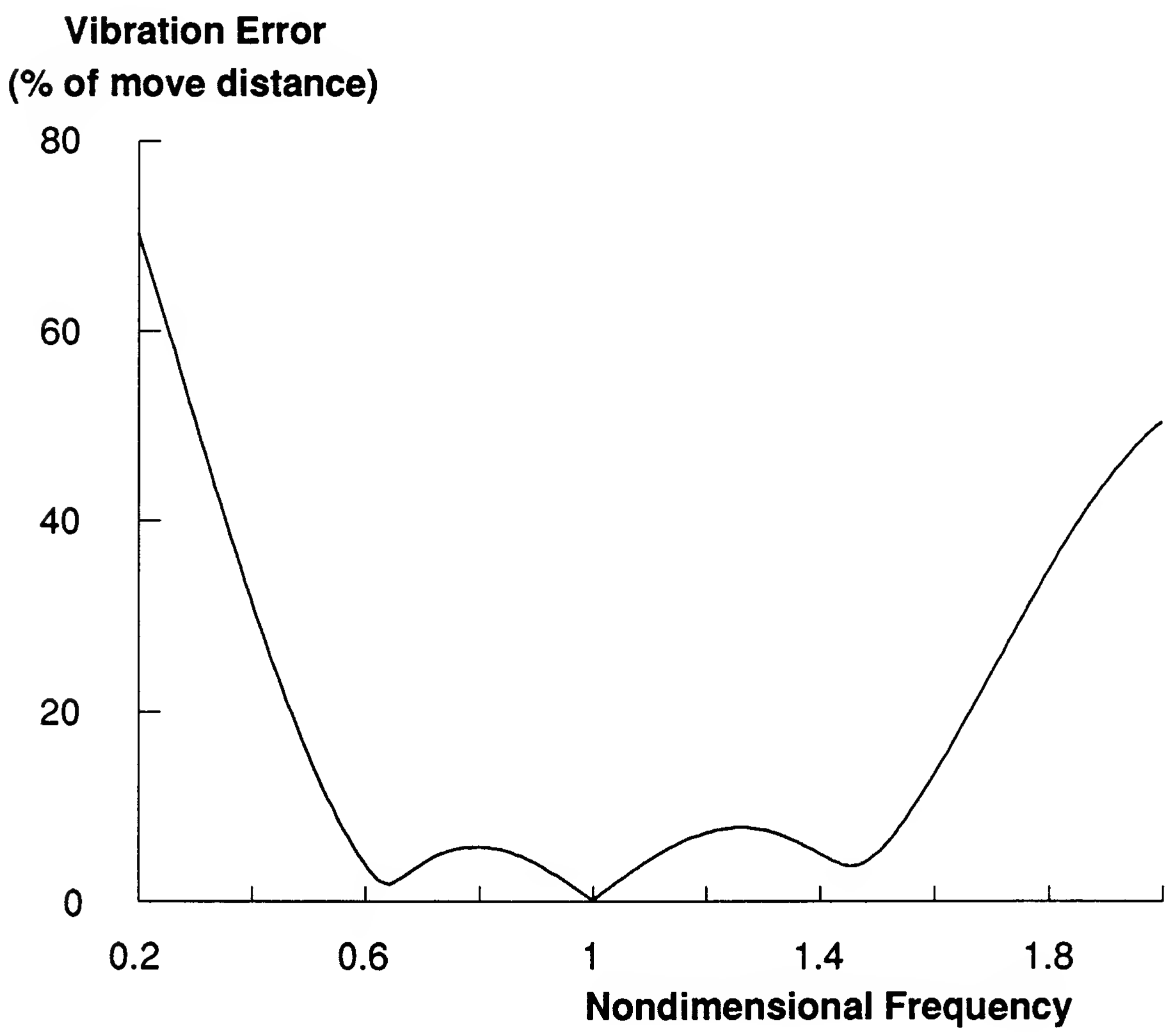


Figure 3.4: Example of using additional robustness constraints to generate an pulse sequence. This is a curve of vibration error vs. nondimensional frequency for a pulse sequence that meets the constraint equations given in this section. Note how the constraint equations force the vibration error to be low over a wide range of frequencies.

tend to be constructed of relatively few pulses. This result becomes important as the phenomenon presented in this chapter is generalized in the next chapter.

Time Domain Technique — Shaping Inputs with Impulse Sequences

Chapter 4

4.1 Introduction

This chapter will generalize the linear programming results presented above. First, the constraint equations will be solved under special conditions without resorting to the linear programming routine. Next, the robustness of these input sequences will be examined. Third, the practicality of these results will be enhanced by extending the results so that they apply to arbitrary inputs. This step involves using the basic system input as a finite impulse response (FIR) filter. Fourth, the filters will be analyzed in several different domains. Lastly, a series of advanced topics and applications will be presented.

4.2 Direct Solution of the Constraint Equations

The derivations of chapter 3 assumed that the input to a system was digital, therefore the input was constructed from a sequence of pulses which were spaced at the digital sampling interval. Under this scenario, the timing of the pulses is fixed and the amplitudes are calculated. This section includes the time of each impulse as a variable in addition to the amplitudes. Because the impulses can now be placed at any arbitrary time, the inputs can be treated as impulses and not digital pulses as in the previous chapter.

First, the two basic constraint equations (equations 3.6) are considered separately. Since there are only two equations, only two unknowns can be solved. Equation 3.6 expressed for the case when only two impulses exist ($N = 2$) results in two unknown quantities:

$$B_1 \cos\phi_1 + B_2 \cos\phi_2 = 0 \tag{4.1}$$

$$B_1 \sin\phi_1 + B_2 \sin\phi_2 = 0 \quad , \tag{4.2}$$

with

$$B_j = \frac{A_j \omega}{\sqrt{1-\zeta^2}} e^{-\zeta\omega(t_{end}-t_j)} \quad ,$$

where A_j is the amplitude of the Nth impulse, t_j is the time of the Nth impulse, and t_{end} is the time at which the sequence ends (time of the last impulse).

Therefore, the values of A_1 and t_1 may be assumed; and the values of A_2 and t_2 may be determined. By selecting 0 for the time of the first impulse (t_1), and 1 for its amplitude (A_1), the two equations with two unknowns (A_2 and t_2) are solved. A_2 scales linearly for other values of A_1. The solution of these two equations is graphically and symbolically given in figure 4.2. The amplitudes of the impulses have been normalized so that they sum to unity. This step is justified because the selection of A_1 was arbitrary. Since the equations are transcendental, there are an infinity of periodic solutions to the

equations. The solutions that are shown in this chapter are the fundamental solutions. They are the shortest time-duration sequences that will satisfy the equations assuming that only positive amplitudes are used. In section 4.8, the shorter, negative amplitude solutions are considered.

Figure 4.1 graphically demonstrates why the two impulse solution achieves a vibrationless response from the system. The two responses shown can be superposed so that the system moves forward without vibration after the input has ended.

4.3 Robustness

4.3.1 Robustness to Errors in Natural Frequency

The two-impulse input, however, cancels vibration only if the system natural frequency and damping ratio are exact. In order to quantify the residual vibration level for a system, a vibration-error expression must be defined, here as the maximum amplitude of the residual vibration during a move as a percentage of the amplitude of the rigid body motion. The system for which this value is computed is a simple harmonic oscillator. No numerator dynamics are included. The vibration error plots shown in this chapter are generalized in section 4.5.5 to more complex systems. Figure 4.3 shows a plot of the vibration error as a function of the system's actual natural frequency. The input was designed for a system with a natural frequency of ω_0. Acceptable response is defined as less than 5% residual vibration for the simple oscillating system. Figure 4.3 shows that the two-impulse input is robust for a frequency variation of less than $\approx \pm 5\%$.

In order to increase the robustness of the input under variations of the system natural frequency, a new constraint may be added. The derivatives of (3.4) and (3.5) with respect to *frequency* (ω) can be set equal to zero — the mathematical equivalent of setting a goal of small changes in vibration error for changes in natural frequency. The resulting

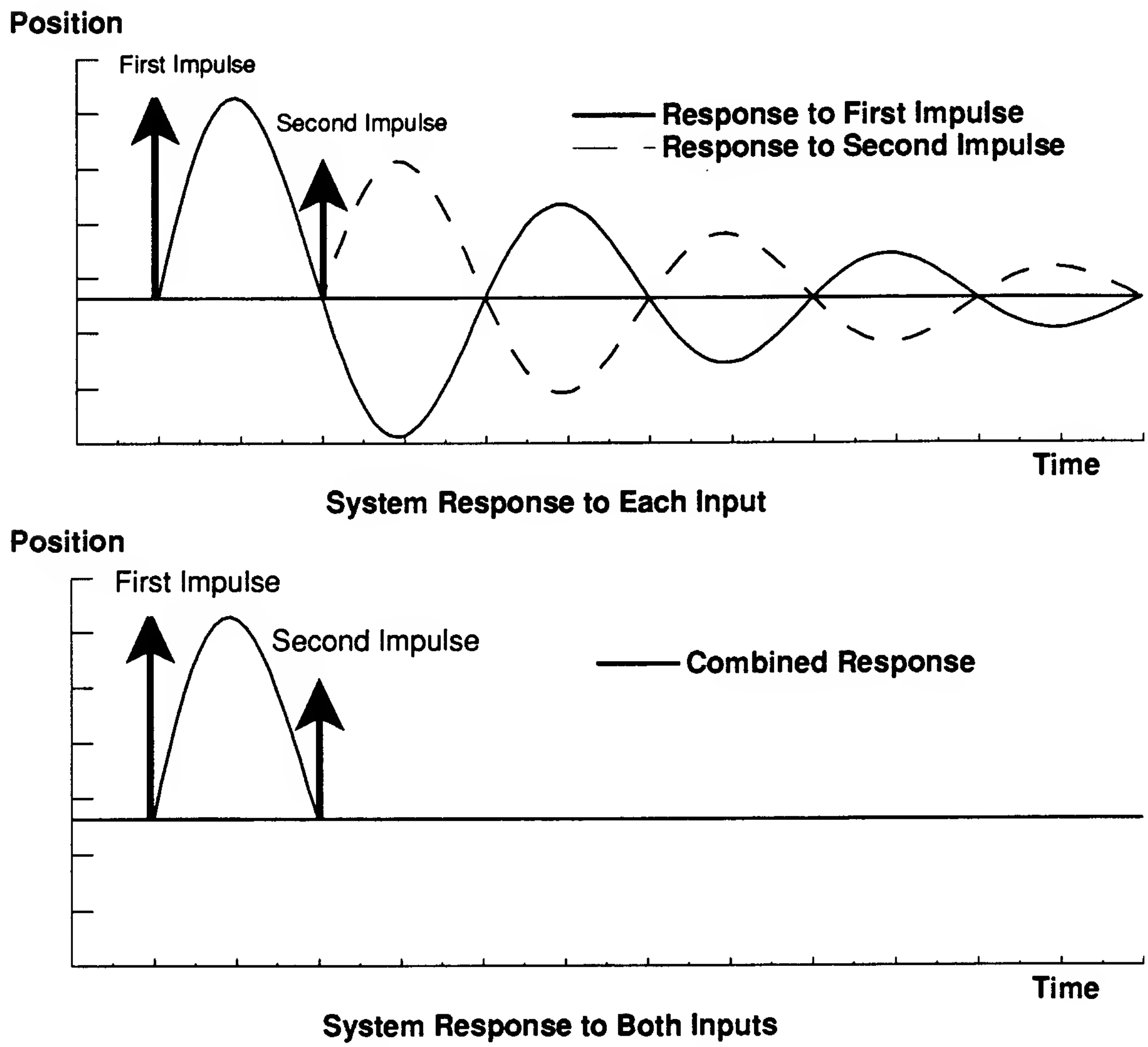
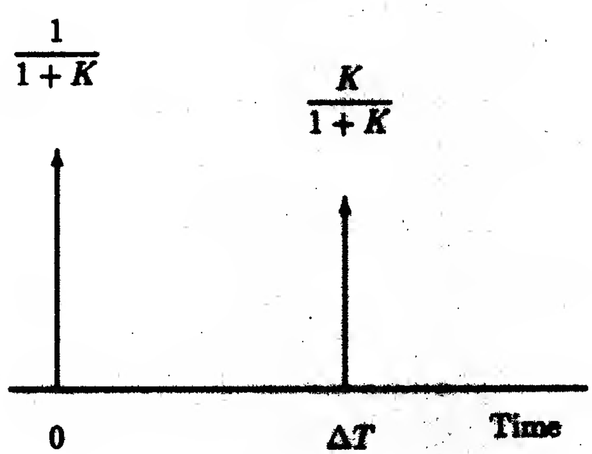


Figure 4.1: Generating a vibrationless output. The two impulse responses shown add to form an output that shows a net positive motion with no vibration after the input has ended. The system moves forward and returns to a zero position — analogous to the impulse command.



$$K = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}$$
$$\Delta T = \frac{\pi}{\omega_0\sqrt{1-\zeta^2}}$$

Figure 4.2: Two-impulse input — designed to have a vibration-error expression which is zero at the expected system natural frequency, ω_0 (radians). ζ is the expected damping ratio.

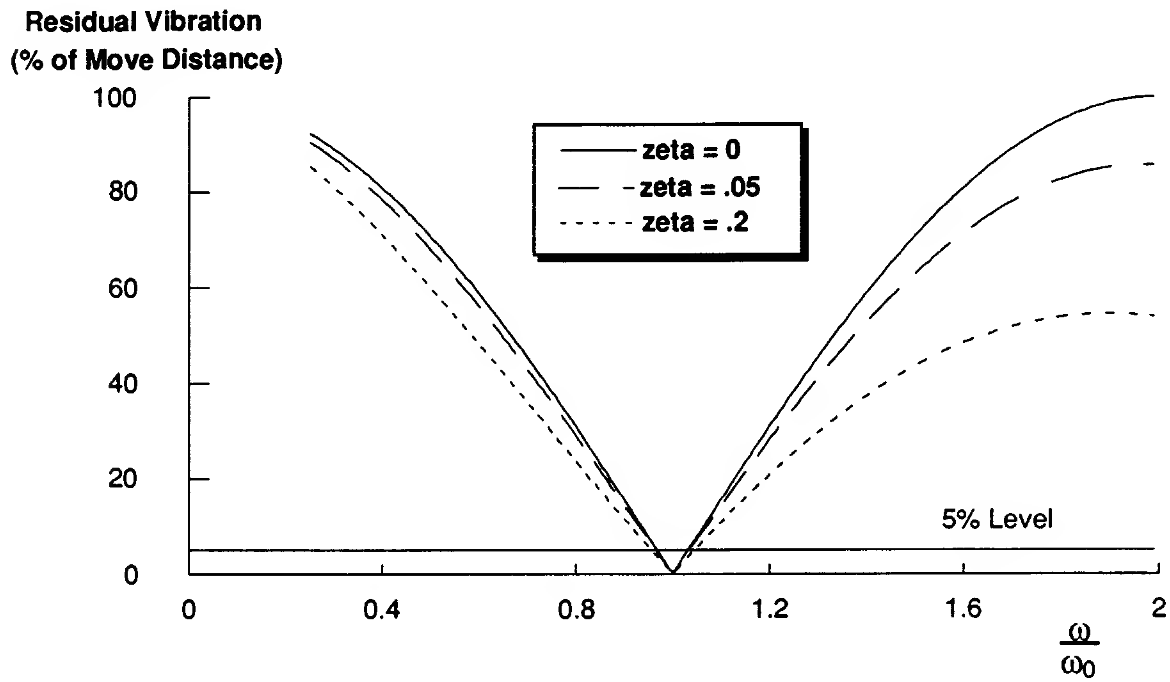


Figure 4.3: Vibration error vs. system natural frequency for three systems with different values of damping ratio excited by the two-impulse sequence in figure 4.2.

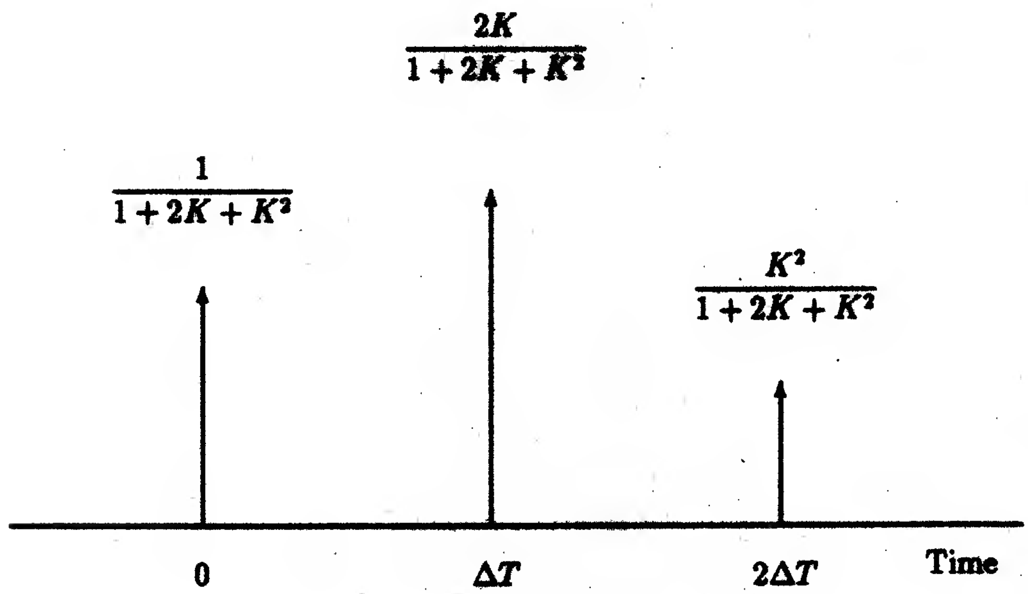
constraints are (see Chapter 3):

$$\begin{aligned}
\sum_{j=1}^{N} A_j(t_j)e^{-\zeta\omega(t_{end}-t_j)} \sin\left(t_j\omega\sqrt{1-\zeta^2}\right) &= 0 \\
\sum_{j=1}^{N} A_j(t_j)e^{-\zeta\omega(t_{end}-t_j)} \cos\left(t_j\omega\sqrt{1-\zeta^2}\right) &= 0
\end{aligned} \tag{4.3}$$

with $\omega = \omega_0$.

Two equations are added to the system; therefore, two more unknowns must be added by increasing the input from two to three impulses (added unknowns: A_3 and t_3). Assuming arbitrary values of $A_1 = 1$ and $t_1 = 0$; solving for the values A_2, A_3, t_2, and t_3; and then normalizing so that $A_1 + A_2 + A_3 = 1$ produces the answers shown in figure 4.4. The corresponding vibration error curves are shown in figure 4.5. In this case, the input is robust for system frequency variations of $\approx \pm 20\%$ Results of experiments on a mechanical system are presented in chapter 5.

The process of adding robustness can be further extended to include the second



$$K = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}$$

$$\Delta T = \frac{\pi}{\omega_0\sqrt{1-\zeta^2}}$$

Figure 4.4: Three-impulse input — designed to have a vibration-error expression which is both zero and tangent at the expected system natural frequency, ω_0 (radians). ζ is the expected damping ratio.

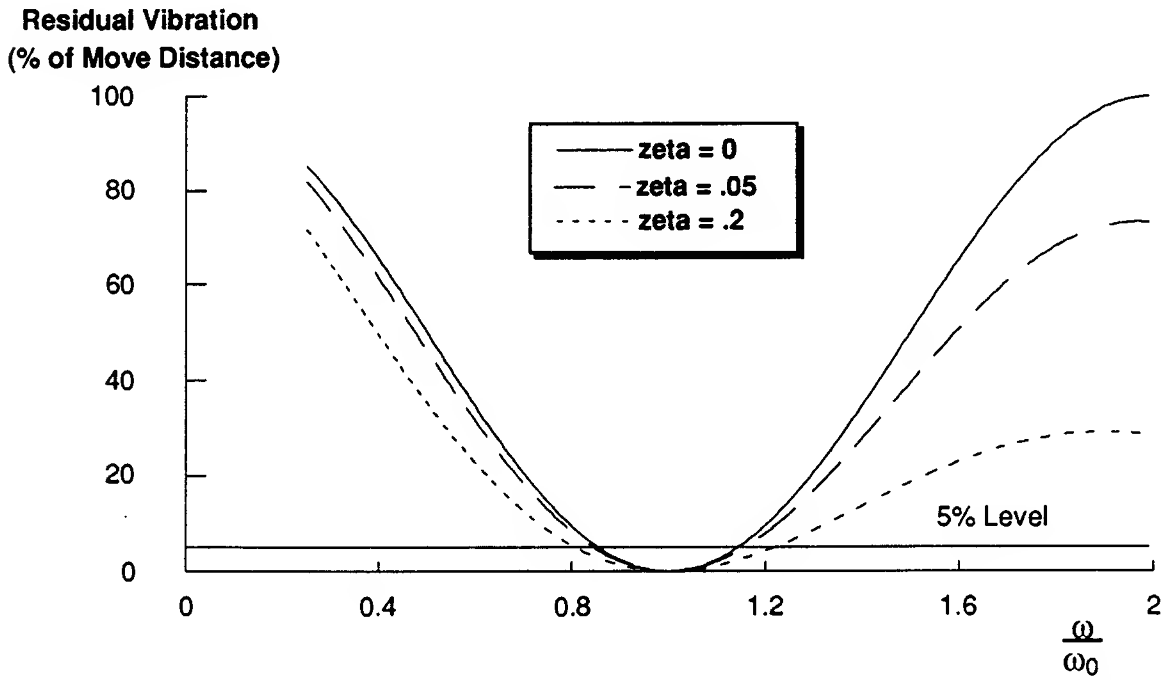
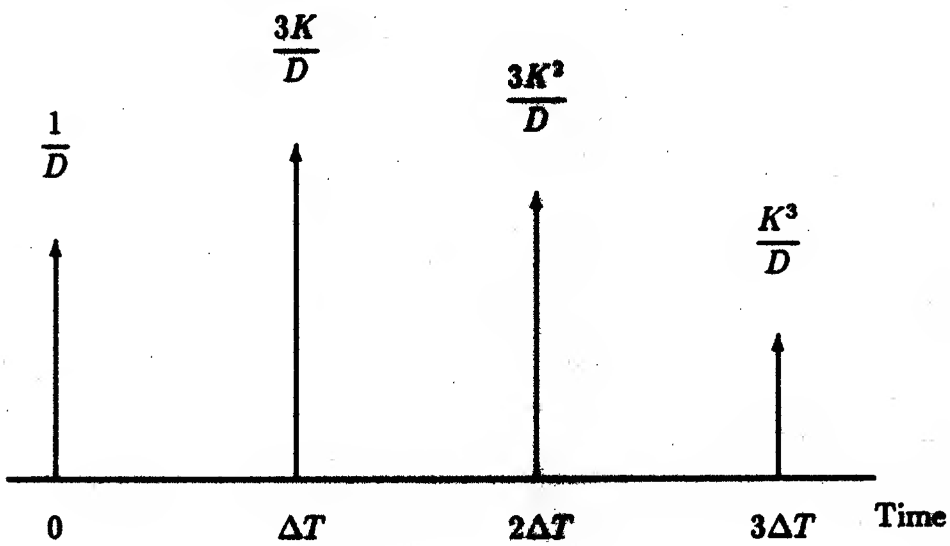


Figure 4.5: Vibration error vs. system natural frequency for three systems with different values of damping ratio excited by the three-impulse sequence in figure 4.4.

derivatives of (3.4) and (3.5) with respect to ω. Setting the second derivatives to zero requires that the vibration error be flat around the intended natural frequency. Two more constraint equations are added, therefore, the impulse sequence is increased by one to a total of four impulses. The corresponding input and vibration error curves are shown in figures 4.6 and 4.7. In this case, the input is robust for system frequency variations of $\approx -30\% + 40\%$.

4.3.2 Robustness to Errors in Damping

In order for these system inputs to be insensitive to system parameter variation, uncertainty in damping ratio must also be considered. As with respect to natural frequency in the previous section, the derivative of the amplitude of vibration with respect to damping ratio (ζ) can be computed. In chapter 3 it was shown that the same expressions that guarantee zero derivatives with respect to frequency also guarantee zero derivatives



$$
\begin{aligned}
K &= e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \\
\Delta T &= \frac{\pi}{\omega_0\sqrt{1-\zeta^2}} \\
D &= 1 + 3K + 3K^2 + K^3
\end{aligned}
$$

Figure 4.6: Four-impulse input — designed to have a vibration-error expression which is zero, tangent, and flat at the expected system natural frequency, ω_0 (radians). ζ is the expected damping ratio.

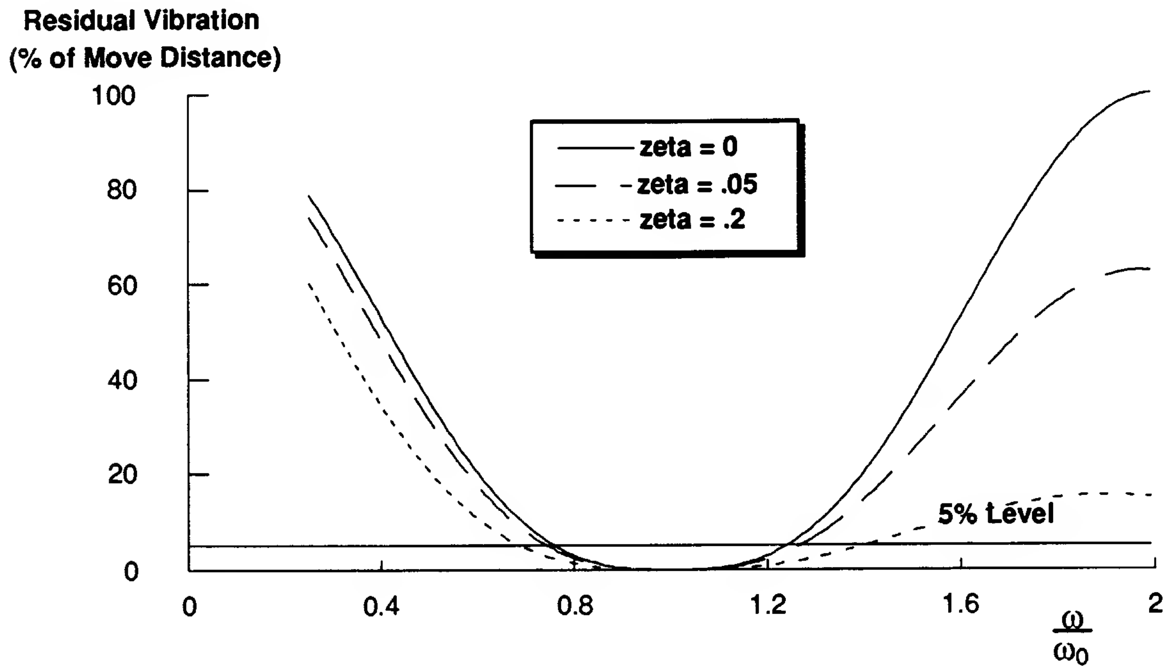


Figure 4.7: Vibration error vs. system natural frequency for three systems with different values of damping ratio excited by the four-impulse sequence in figure 4.6.

with respect to damping ratio. Therefore, robustness to errors in damping has already been achieved by the addition of robustness to errors in frequency. Figure 4.8 shows the vibration-error expression as a function of changes in damping ratio for the same three sequences as were generated in 4.3.1. The exact system damping ratio is .05 in this example. Note that extremely large variations in damping are tolerated. Changes in damping do not have as large an effect on the vibration error as changes in frequency. This fact is fortunate since it is generally easier to measure frequency than it is to measure damping.

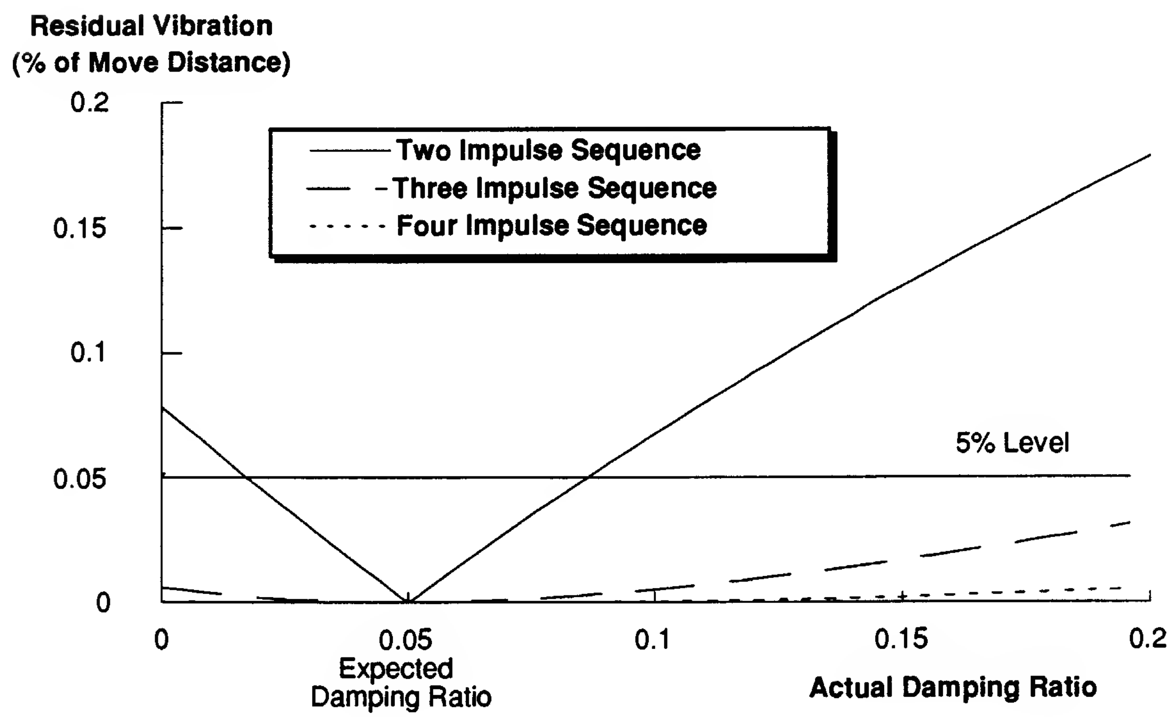


Figure 4.8: Vibration error vs. damping ratio for the two, three, and four-impulse inputs presented in section 4.3.1 calculated for a system with a damping ratio of .05.

4.4 Using Impulse Sequences to Determine System Commands

Sections 4.2 and 4.3 have presented a method for obtaining the shortest system impulse input sequence (constructed of only positive impulses) which simultaneously eliminates vibration at the natural frequencies of interest and includes a specified degree of robustness to system variability. This statement is true because the equations 3.4, 3.5, and 4.3 have no solutions for a smaller t_3 **assuming that the A_i are positive quantities.** This section presents a method for using these "time-optimal" sequences to generate arbitrary inputs with the same vibration-reducing properties.

Systems can not be commanded with impulses as inputs. Therefore, these "input sequences" are merely a conceptual system input. They represent the shortest inputs that meet the desired design criteria. Therefore, if the system were to be commanded to make an impulse motion, the best that could be commanded in order to achieve a vibration-reducing move is the multiple-impulse sequence that was derived for the system. Since all complex moves are just superpositions of elemental moves, complex vibration-reducing moves can be constructed from elemental vibration-reducing moves. Just as the single impulse is the building block from which any arbitrary function can be formed, the impulse sequence can be used as a building block for arbitrary vibration-reducing inputs. This superposition is accomplished by convolving any arbitrary desired input to the system with the impulse sequence in order to yield the shortest actual system input that makes the same motion without vibration. The sequence, therefore, becomes a prefilter for any input to be given to the system. The time penalty resulting from prefiltering the input equals the length of the impulse sequence (on the order of one cycle of vibration for the sequences shown in 4.3). Figure 4.9 shows the convolution of an input (for example, the signal from a joystick in a teleoperated system) with a

non-robust, two-impulse sequence.

Note that actual impulse sequences are **never** sent to the system unless the requested motion is an impulse. The filtered system input is always as smooth as the requested motion. The convolution process is essentially just repeating the requested motion several times.

The impulse sequences from 4.3 have been normalized to sum to one. This normalization guarantees that the convolved motor inputs never exceed the maximum value of the commanded input. If the commanded input is completely known in advance for a particular move, the convolved motor input can be rescaled so that the maximum value of the function is the actuator limit of the system. It should be noted that this form of shaping will work on any type of input — torque, velocity, position, etc. because these quantities are related by integration. The integral or derivative of a vibrationless signal is also vibrationless.

If the commanded input results in system saturation, the shaped command will also saturate the system. Under these conditions, the vibration reduction will not work. A peak value for the command must be selected and never exceeded before the command is shaped. This enables the vibration reduction to perform as expected.

For much of the analysis impulses and pulses were used in input sequences interchangeably. The justification of this can now be provided. If an impulse sequence is convolved with a unit pulse (whose width is determined by the sampling rate or some other constraint), the result is a sequence of pulses. The leading edge of each pulse is a the spacing of the impulses and the heights are scaled accordingly. We have shown above that the convolution of any signal with some impulse sequence results in a signal that has the same vibration-reducing effects as the impulse sequence. Therefore, the pulse sequence will be vibration-reducing. In order to simplify the analysis of a pulse sequence, the equivalent impulse sequence (with each impulse beginning of the leading edge of each pulse) can be used.

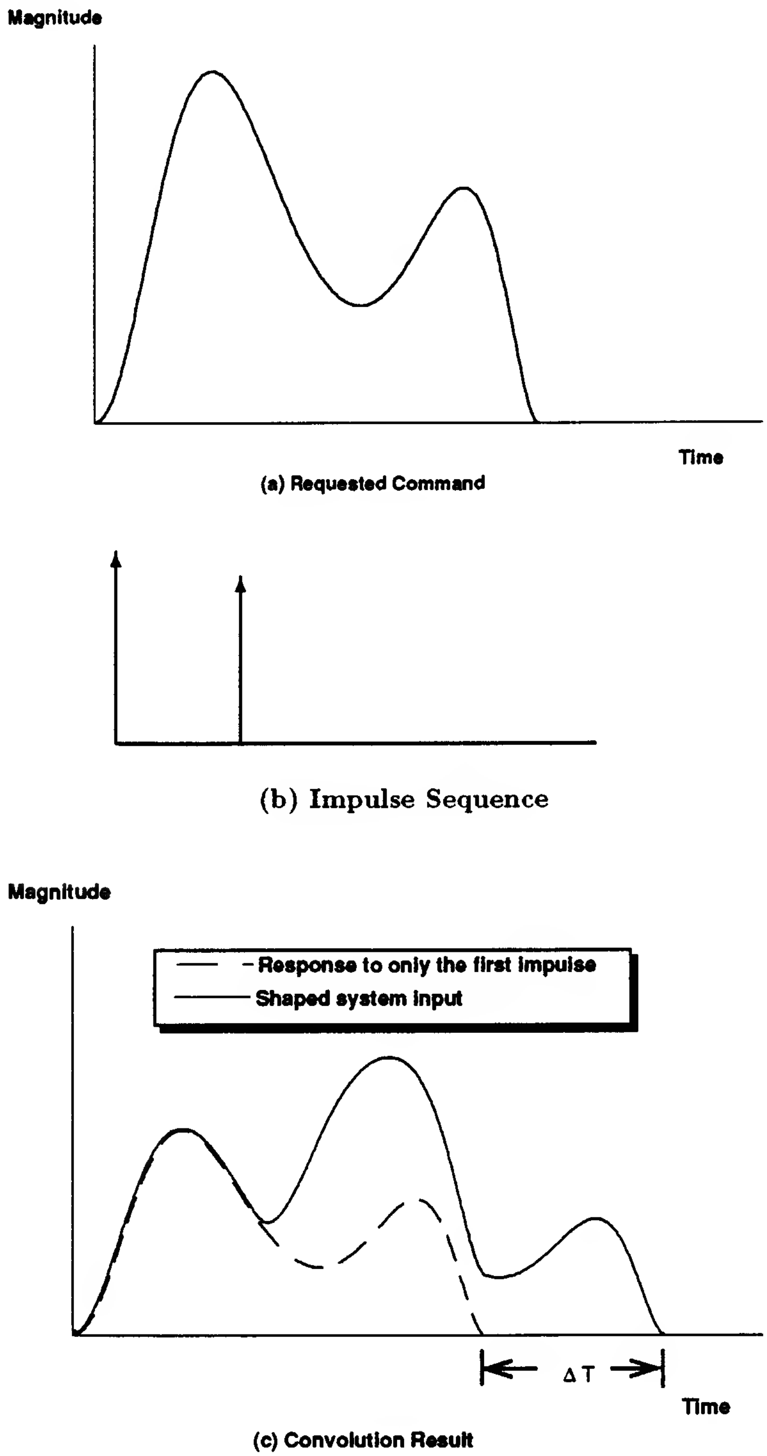


Figure 4.9: Shaping arbitrary inputs. Shown is the convolution of a command in (a) with a two-impulse sequence shown in (b) yielding the system input shown in (c).

For historical reference, the result of convolving a non-robust two-impulse sequence with a step input yields the Posicast input developed by O.J.M. Smith in 1958 [127]. The robustness plot of figure 4.3 demonstrates why Posicast is not generally used.

4.5 Evaluation of Filtering using Impulse Sequences

The concept of filtering with impulse sequences has been presented in the previous section. The goal of this section is to present the concept from several different points of view in order to put this work in proper perspective. The vibration cancellation and robustness effects will be presented in five different representations. First, a new representation, called vector diagrams will be presented. Second, a time domain presentation will be discussed. Third, these filters will be examined in the frequency domain. Fourth, a phase plane analysis will be presented. Fifth, the effect of shaping inputs using impulse sequences will be discussed in terms of transfer function analysis.

4.5.1 Vector Diagrams

A new representation which graphically demonstrates how the new shaping technique works was developed for analysis and impulse sequence-generation purposes. This section will present this new representation which we will call vector diagrams.

The concept of shaping inputs using impulse sequences is based on time domain cancellation of system vibrations. If an impulse is input into the system, it causes an oscillation with a particular phase. If the same input is repeated later in time, the same response is generated. By delaying the second impulse input, the relative phase of the responses is determined. The phasing of the oscillatory responses was demonstrated in figure 4.1.

Figure 4.10 shows that the impulses in time can be plotted as vectors around a circle as an alternative to time line plots. The first impulse (by arbitrary convention) is plotted

in the positive x direction, its angle is 0 (corresponding to zero time) and its length is the amplitude of the impulse, A. The other impulses are then also plotted as vectors in this new space. The angle of the vector is $\phi_i = \omega_g \delta T_i$. The length of the vectors are their respective amplitudes. The frequency, ω_g is arbitrary. Any value can be selected and a new graph will be generated. By making the frequency, ω_g, equal to a natural frequency of the system, the graph becomes a useful tool for determining the vibration reducing properties of an impulse input sequence.

If ω_g is a natural frequency of the system, then the impulses can represent the amplitude of an oscillation at this frequency, the resultant vector of the set of impulses plotted in the vector diagram represents the residual vibration after the last impulse has been input into the system. The magnitude of the resultant vector is the amplitude of the oscillation, the angle of the vector is the phase of the oscillation (where a phase of zero means in phase with the oscillation induced by the first impulse). This fact can be proven by noting that the equations of the vector resultant:

$$A_1 \cos\phi_1 + A_2 \cos\phi_2 + \cdots + A_N \cos\phi_N = 0 \tag{4.4}$$

$$A_1 \sin\phi_1 + A_2 \sin\phi_2 + \cdots + A_N \sin\phi_N = 0 \quad , \tag{4.5}$$

are the same as those in equation 3.4 and 3.5 when the mode of interest has no damping. See subsection 4.5.1 which explains the inclusion of system damping.

Using Vector Diagrams for Analysis

Figure 4.11 shows that the two impulse sequence exactly cancels vibration when the frequency is exactly as anticipated when the sequence was designed. The lower portion of the figure demonstrates the effect of designing the sequence for one frequency while the oscillations are occurring at a nearby frequency. Note that for small frequency perturbations, the vectors still cancel to a first order in the x-direction. However, a significant resultant (which is a direct measure of residual vibration) appears in the

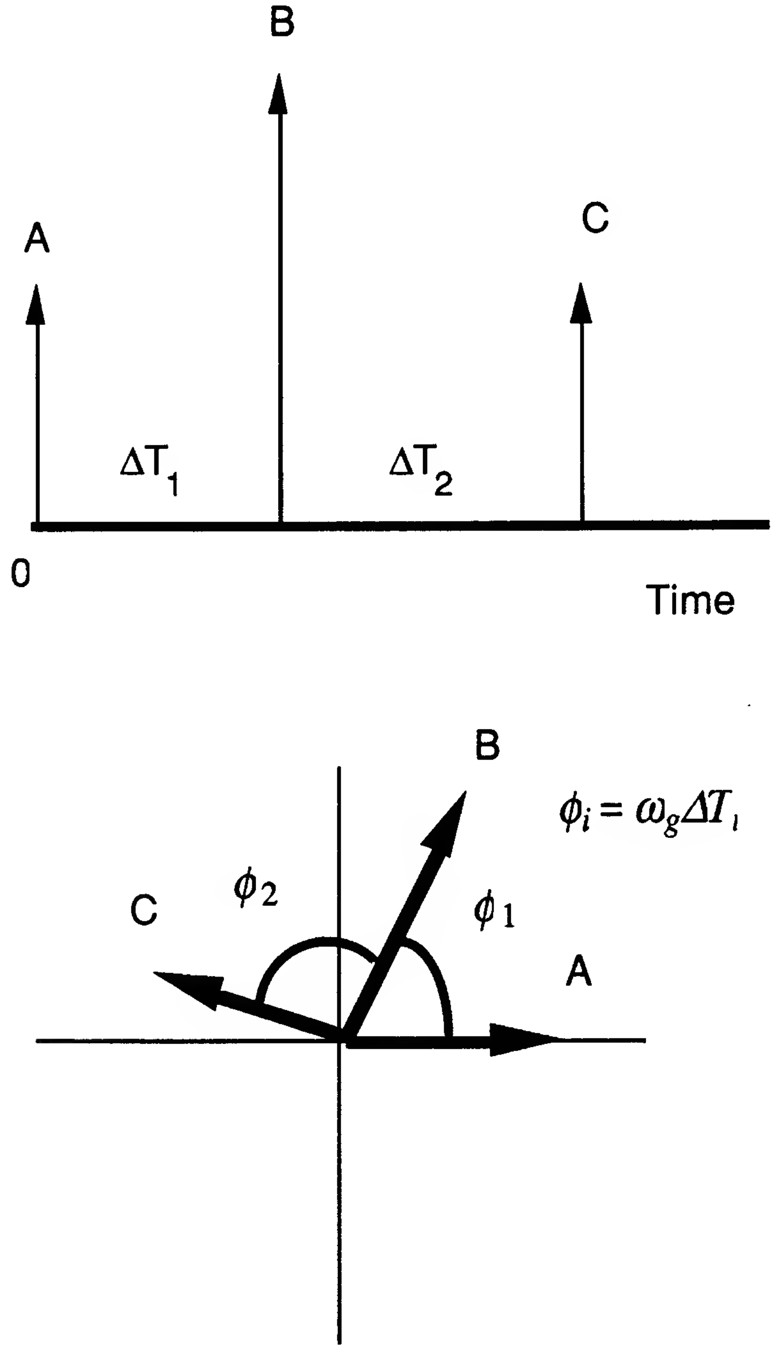


Figure 4.10: Creating a vector diagram. The impulses of a sequence can be plotted as vectors in a cylindrical space. ω_g is a frequency of interest (for vibration reduction, the natural frequency is used).

y-direction. This graphically demonstrates the lack of robustness of the two impulse sequence.

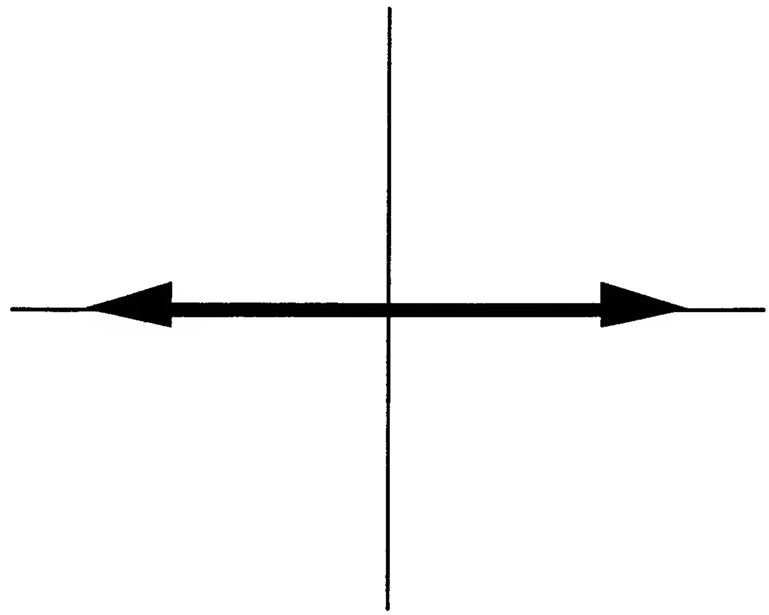
Figure 4.12 shows the vector diagram for a three impulse sequence. When the frequency is exactly as expected, the vectors cancel exactly. However, the robustness of this sequence can be graphically demonstrated by plotting the vector graph with an error in frequency. Note that the vectors cancel in both the x and y-directions to a first order approximation .

Figure 4.13 demonstrates the same effect for the four impulse sequence. For small errors, the angle θ is small. Using the polynomial expansion for sine and cosine of θ, all terms in θ and θ^2 cancel. This sequence of figures shows why the impulse sequences become more robust to frequency shifts or uncertainties as the higher derivative constraints are included.

Now that the properties that create robustness have been demonstrated, the vector diagrams can be used as a tool for generating new sequences. Any new sequence can be plotted on a vector graph. The order to which its vectors cancel gives the level of robustness for the sequence. For example, any sequence that cancels to a second order is functionally equivalent to the four-impulse sequence. All of the vibration data that is developed for the four impulse sequence applies to this new sequence.

Vector Diagrams With Damping

When the mode is damped, two slight modifications must be made to the vector diagrams. First, damping alters the frequency by the factor $\sqrt{1-\zeta^2}$. The damped natural frequency is the one which is used to plot the vectors since the oscillations from the system will be at this frequency. Second, the amplitudes of the vectors must shrink with time in order to cancel. Figure 4.14 shows a damping spiral superimposed over the vector diagram. Fortunately, the effect of damping on the vector magnitudes can be initially ignored. The vectors are plotted using the damped natural frequency but the



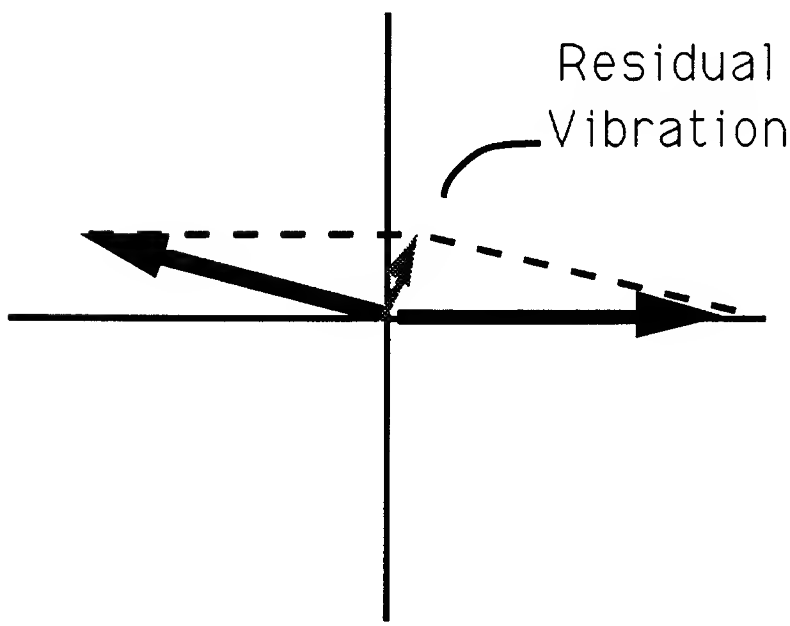


Figure 4.11: The two impulse sequence shown in a vector diagram. In both plots the chosen frequency, ω_g, is exactly the natural frequency of a system that needs to be controlled. The top plot is the case when the design frequency of the sequence matches the natural frequency of the system. The bottom plot is the case when the design frequency of the sequence does not match the natural frequency of the system.

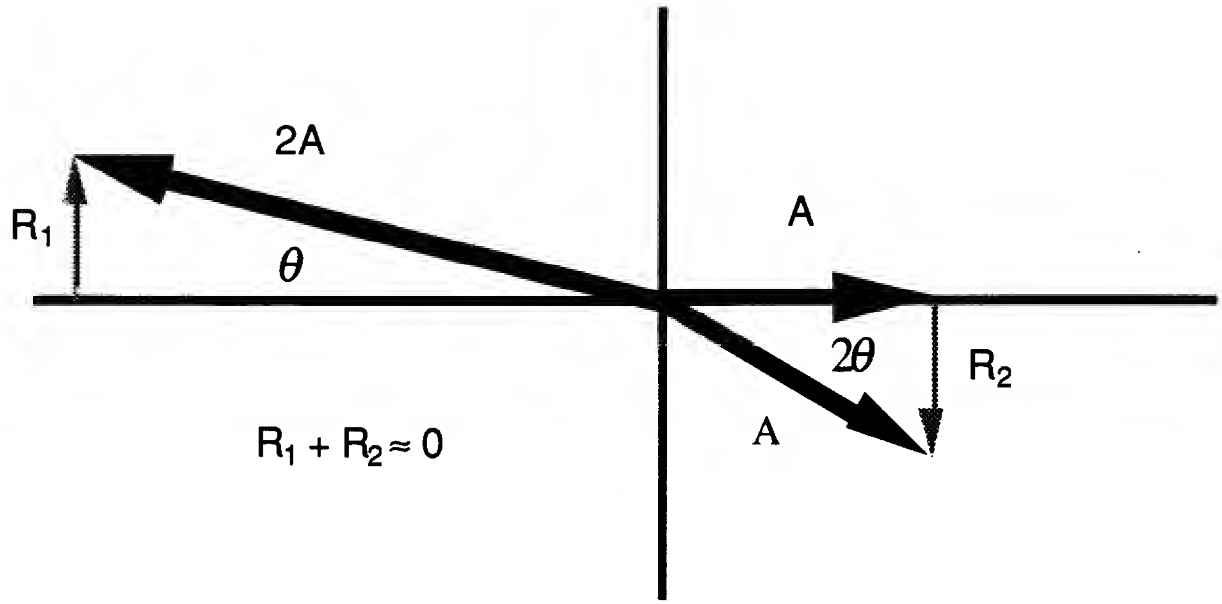


Figure 4.12: The three impulse sequence shown in a vector diagram. The chosen frequency, ω_g, is exactly the natural frequency of a system that needs to be controlled. Shown is the case when the design frequency of the sequence does not match the natural frequency of the system.

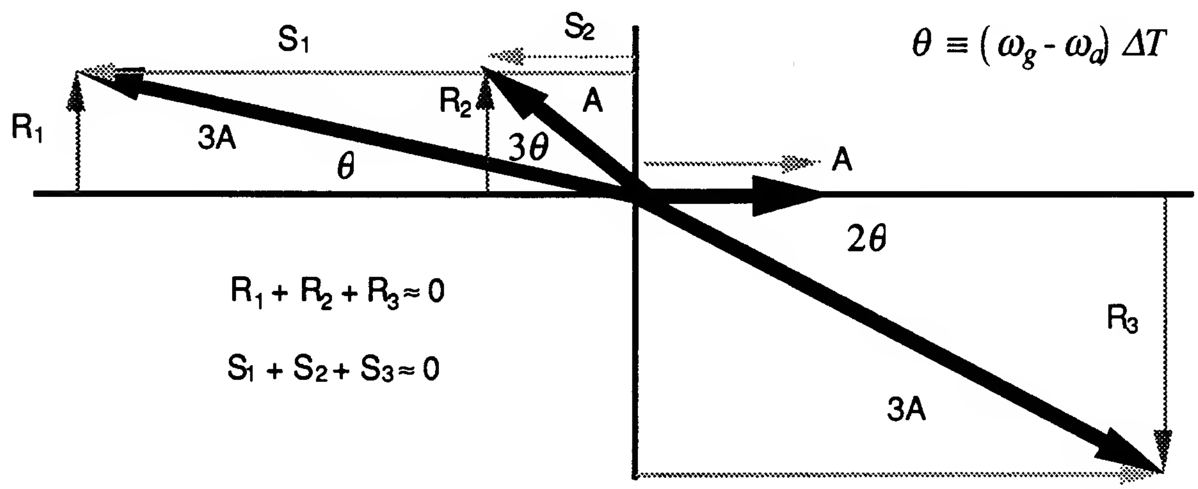


Figure 4.13: The four impulse sequence shown in a vector diagram. Shown is the case when the design frequency of the sequence (ω_g) does not match the natural frequency of the system (ω_a). The vector diagram is drawn so that 2π radians corresponds to one period of ω_a, the **actual** frequency of the system.

magnitude change is not included. Once the analysis is complete, the damping spiral can be superimposed on the vectors to get their final magnitude. For all of the analyses, the undamped magnitudes will be used to simplify the derivation of sequences.

One powerful feature of the vector diagrams is that they automatically nondimensionalize the problem of impulse sequence generation. One revolution is the natural frequency of the system. If a sequence is developed for one natural frequency, it can be easily changed for a new system. The angles of the vectors remain the same — only the actual mapping into impulse times changes. (Remember that the angles are related to time by $\theta = \omega_g \Delta T$.) Additionally, the damping effect can be separated from the vector cancellation. A basic sequence can be generated for a system assuming that it has no damping. After a sequence is selected, the damping spiral can be superposed as the last step before implementation. An actual implementation would follow the following steps:

- Select a sequence for an undamped system with the same natural frequency as the system of interest (ω_g).
- Determine the impulse times using the damped natural frequency ($\omega_g\sqrt{1-\zeta^2}$).
- Scale the vectors by the damping spiral (multiply each vector at time, t, by $e^{-\zeta\omega_g t}$)

4.5.2 Time domain Analysis

This section explains the relationship between the robustness of various impulse sequences. Figure 4.15 demonstrates pictorially how a pair of two-impulse sequences each yields a small residual vibration because of errors in estimating the natural frequency of the system. These two residuals have the same amplitude (R_0 in this example) and phase (relative to the first of the two impulses). The residual amplitude (R_1 in this example) is the same for any set of vectors A and B with the same time separation, ΔT (the assumption of linearity). By placing two pairs of these impulses approximately out

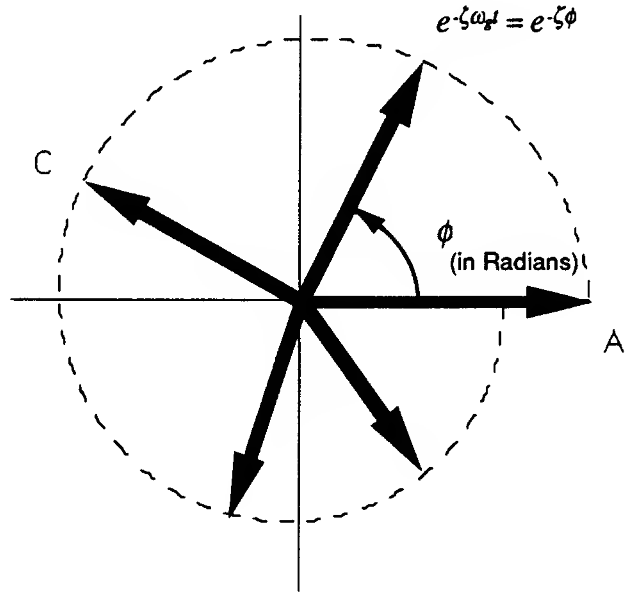


Figure 4.14: Damping spiral shown on a vector diagram. All the vectors on this plot have the same effective magnitude because of system damping. The vibration amplitude caused by an early vector, decays with time. In order to cancel this decaying vibration, a smaller amplitude input must be given to the system. For example the vector, A, causes a vibration that decays to amplitude, C, in T_c seconds. The vector that is equivalent to vector A at a later time, T_c, is vector C. This is equivalent to saying that at times later than T_c, the same amplitude of vibration caused by vector A at $t = 0$ can be caused by vector C at $t = T_c$. The dashed spiral shows by what amount the amplitude must be scaled. The damping spiral is essentially a plot of $e^{-\zeta\phi}$ in the polar coordinates of the vector diagram.

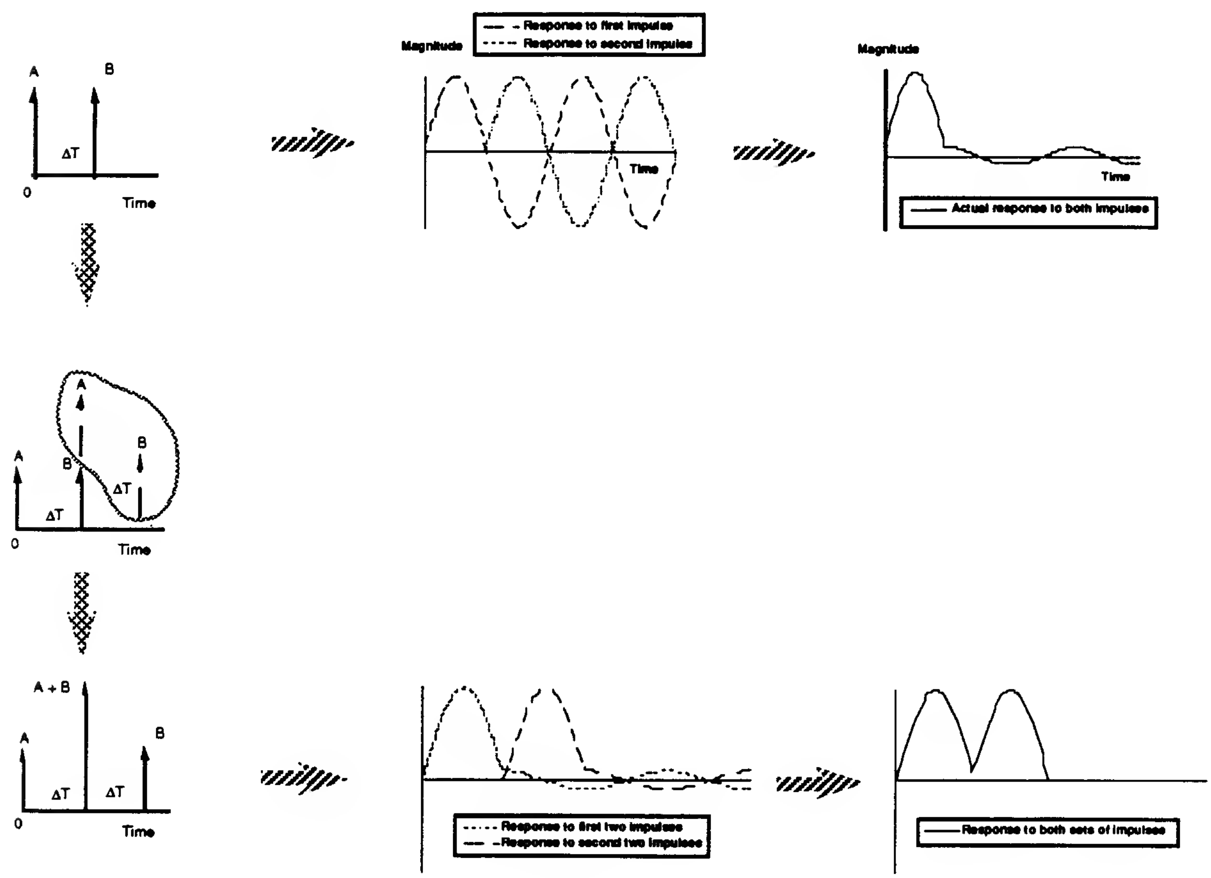


Figure 4.15: Time domain summary. The two impulses in the top half of the figure attempt to cancel vibration but produce a residual (R_1). By constructing a three-impulse sequence when two, two-impulse sequences are summed out of phase, the residuals, R_1, cancel thus producing an even smaller residual, R_2. This is the source of the robustness of the three-impulse sequences.

of phase with each other, as in figure 4.15c, the two residuals will be approximately out of phase. (They are phased by ΔT which is an approximation of the half-period of the oscillation). The two residuals therefore cancel, producing a much smaller residual (R_2). By summing the two pairs of impulses, the three impulse sequence is obtained.

The same effect can be seen in generating the four impulse sequence. Two three-impulse sequences each produce a small residual. Because they are identical inputs, the residual vibration that they produce are equal in magnitude and phase. The two three-impulse sequences are roughly out of phase so an even smaller residual is produced.

4.5.3 Frequency Domain

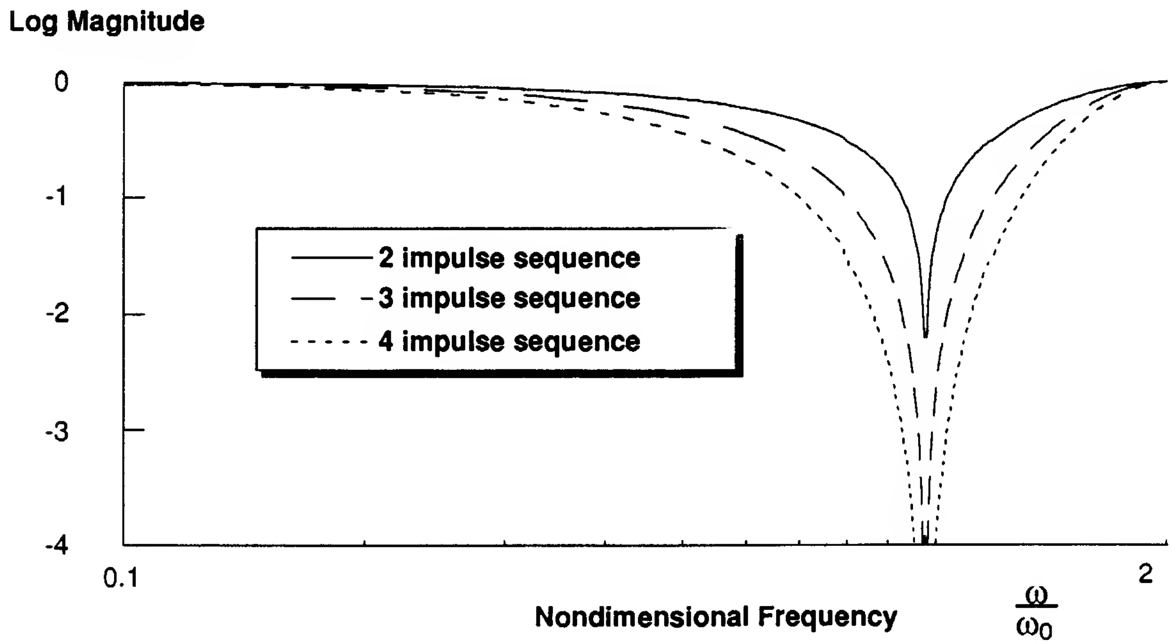
The new type of filter that was developed can also be analyzed in the frequency domain. Figure 4.16 shows the exact Fourier transforms of two, three, and four-impulse sequences for a system that has no damping. Note that the magnitude response must be exactly zero at the natural frequency in order to cancel the infinite magnitude response of a system without damping. The difference between the three sequences is the width of the trough. The four-impulse sequence is more robust due to its wider notch.

Figure 4.17 shows the exact Fourier transforms of two, three, and four-impulse sequences for a system that has .05 damping. Note that the response never drops to zero magnitude — it drops low enough to eliminate the peaks of the system. Figure 4.18 shows the exact Fourier transforms of two, three, and four-impulse sequences for a system that has .2 damping. All of the filters have an infinite number of zeros. This can be seen in the frequency domain as a repetition of the notch at higher frequencies. The infinite number of zeroes is caused by the transportation lag associated with delaying the impulses in the sequences [104] (p. 350). The sequences have no poles. This is important because filters that have poles will be shown in the next chapter to have inferior performance to filters without poles.

The success of the new input shaping (or filtering) technique appears in the frequency domain as a notch that matches the resonant characteristics of the system. As the damping of the system is increased, the notch widens and decreases in depth in order to match the widening and decreasing resonant peak of the system.

4.5.4 Phase Plane

The vibration canceling and robustness can also be observed in the phase plane. Figure 4.19 shows exact cancellation in the phase plane. A vibration is induced by inputting an impulse in position set point. The system is allowed to vibrate through one half cycle



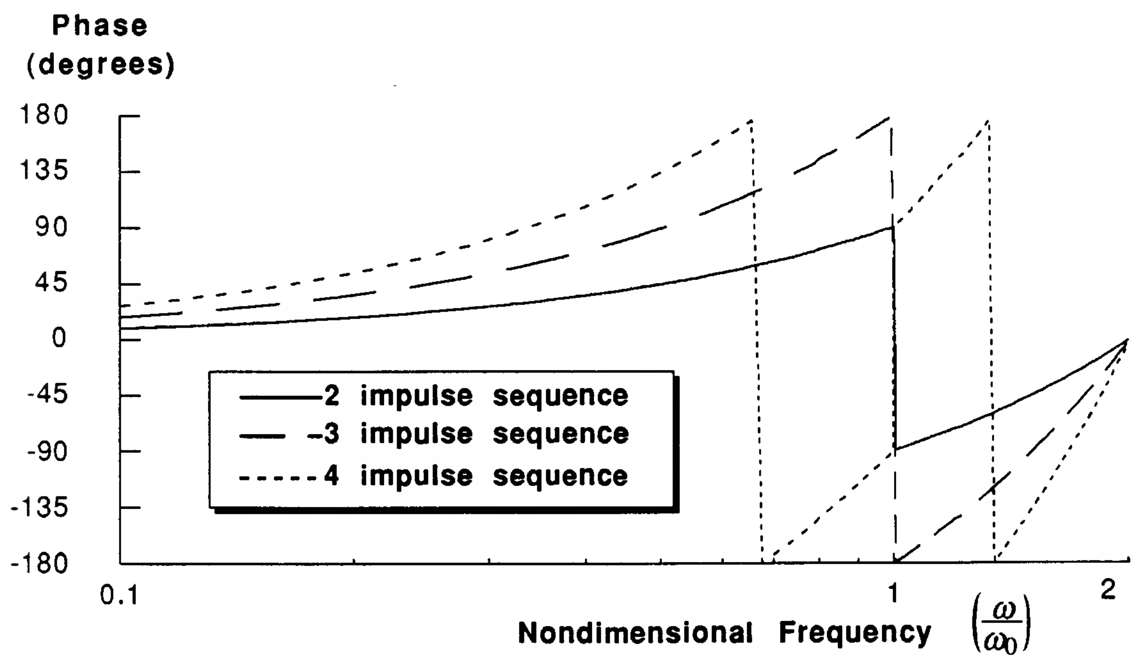


Figure 4.1: Comparison of the two, three, and four-impulse sequence exact Fourier transforms without damping. The frequency is nondimensionalized to ω_0, the anticipated frequency of the system of interest.

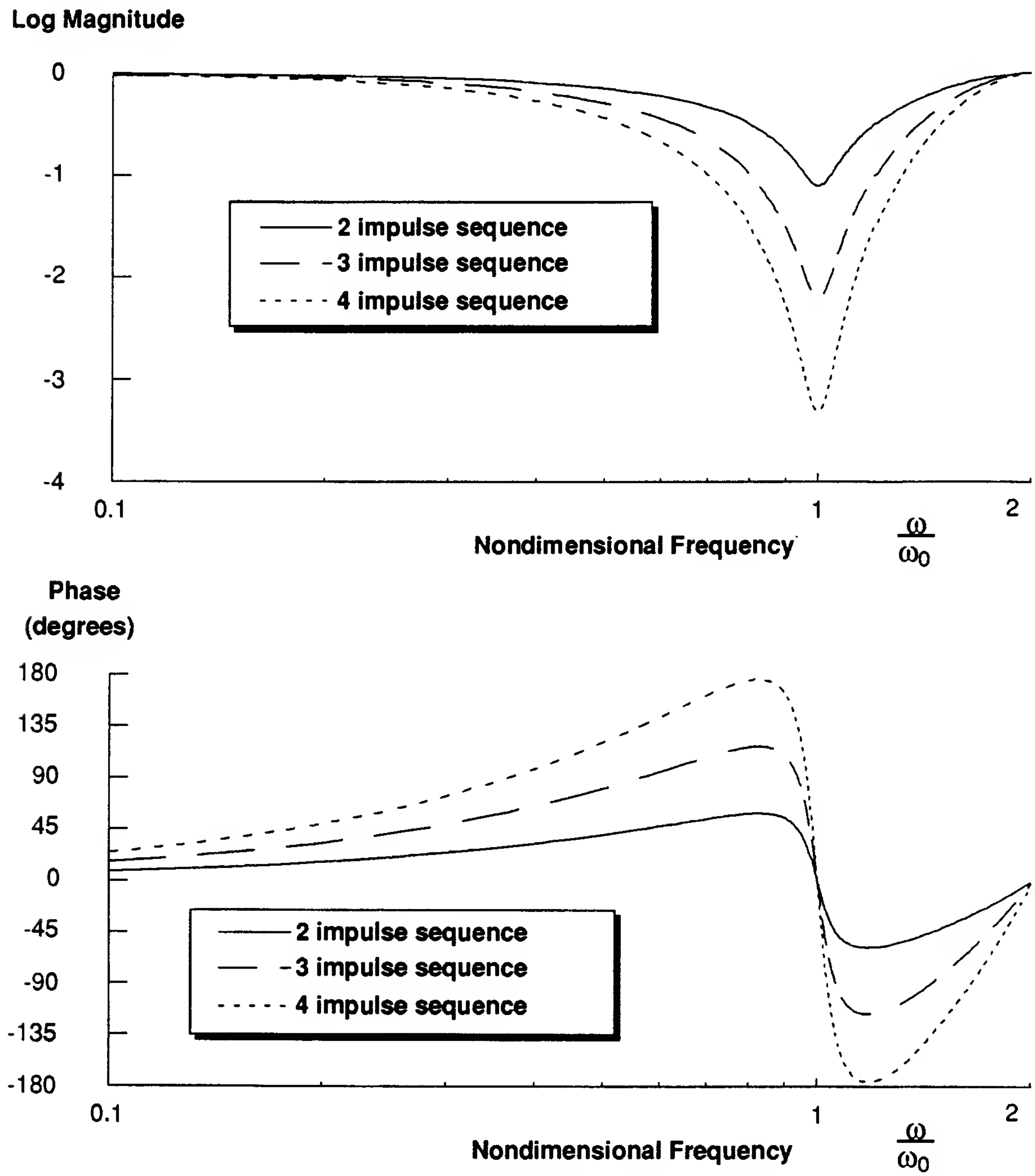


Figure 4.17: Comparison of the two, three, and four-impulse sequence exact Fourier transforms with .05 damping. The frequency is nondimensionalized to ω_0, the anticipated frequency of the system of interest.

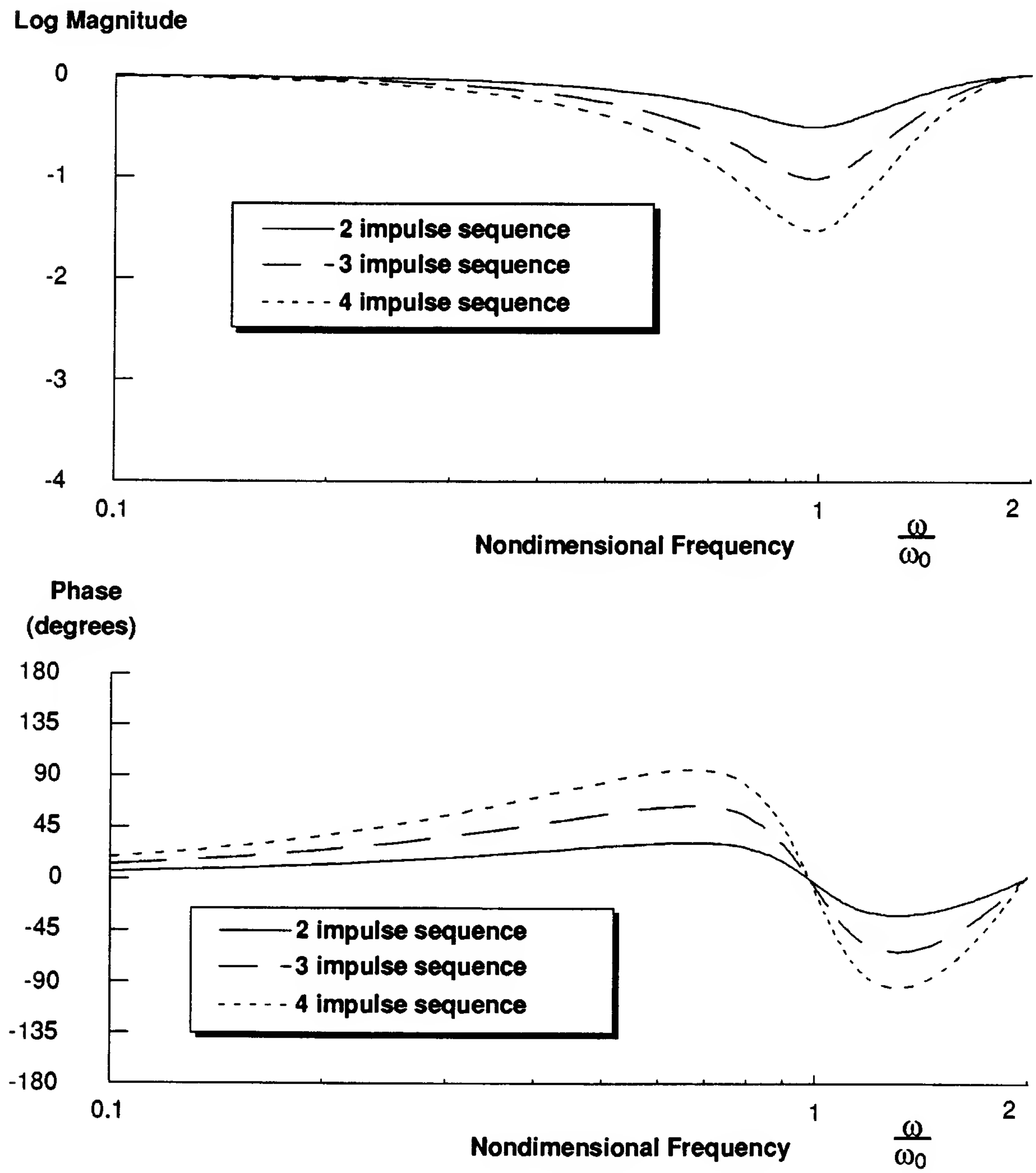


Figure 4.18: Comparison of the two, three, and four-impulse sequence exact Fourier transforms with .2 damping. The frequency is nondimensionalized to ω_0, the anticipated frequency of the system of interest.

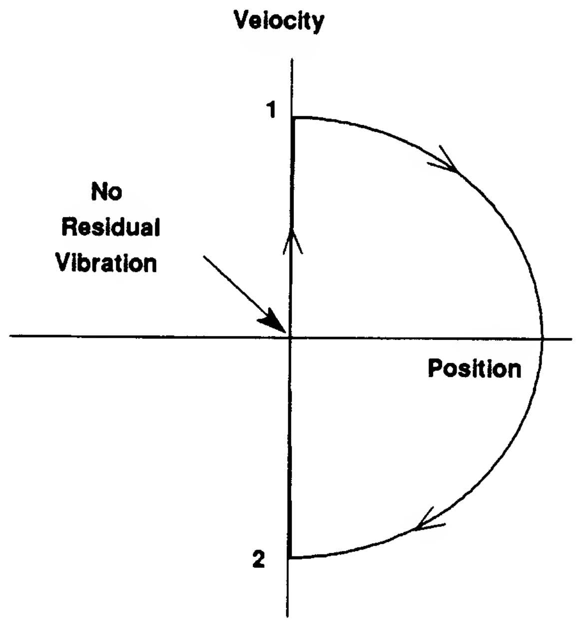


Figure 4.19: Exact cancellation in the phase plane. This is a phase plane plot for a system without damping that was given a two-impulse input. The input was tuned exactly to the system. Note the absence of residual vibration.

until it is has no velocity and then a second impulse setpoint is input to stop the system from vibrating. Because the system is commanded with impulses, it does not actually move to a new location. This was done to simplify the phase plane plot. If the system responds to impulses without vibration, then the system must respond to steps and other trajectories without vibration.

Figure 4.20 shows the effect of an error in frequency estimation. The system has a finite velocity which will result in an oscillation. Figure 4.21 shows a phase plane plot of a robust three-impulse sequence. Note that a frequency underestimate causes the system to overshoot in position at position 2. However, when the system moves from position 3 to position 4 the same underestimate tends to cancel the original overshoot.

Figure 4.22 shows the phase plot for the four-impulse sequence. The same tendency to cancel errors can be seen in this figure. The cancellation happens twice, therefore, the residual vibration is greatly reduced over that of the three-impulse case.

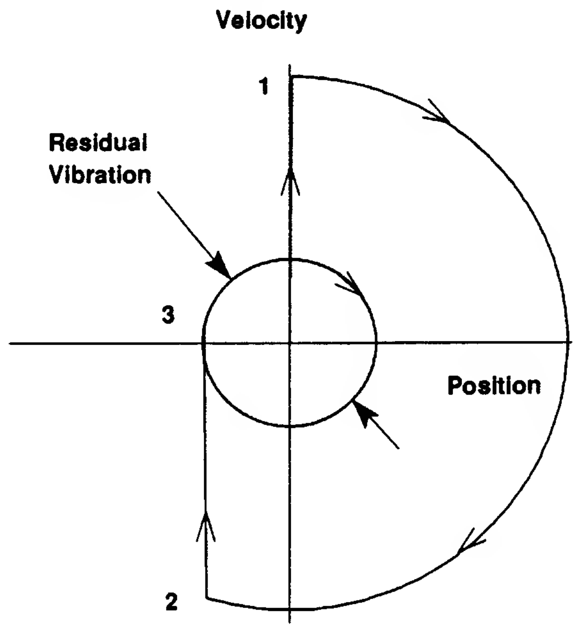


Figure 4.20: Phase plane plot for a two-impulse input. The system's resonant frequency is 10% higher than the anticipated natural frequency.

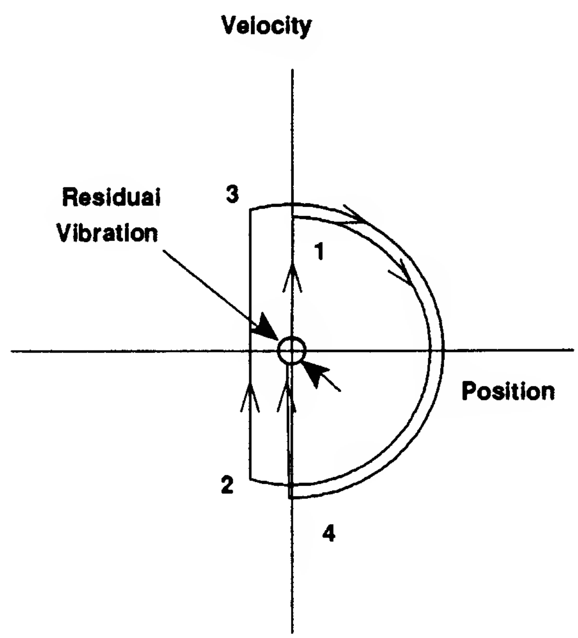


Figure 4.21: Phase plane plot for a three-impulse input. The system's resonant frequency is 10% higher than the anticipated natural frequency.

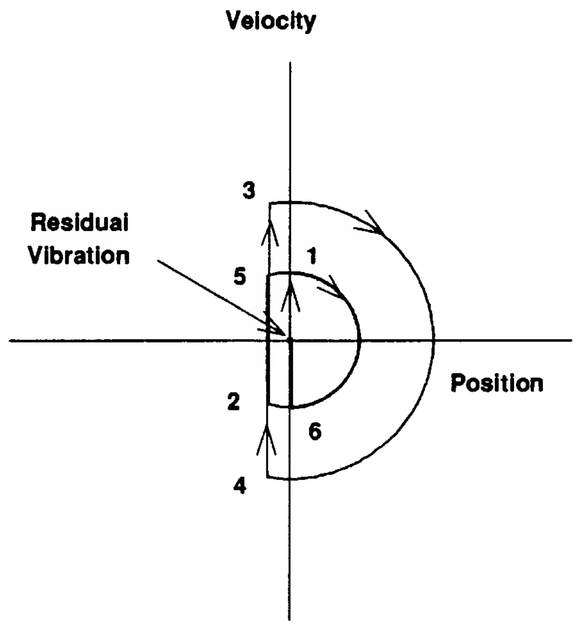


Figure 4.22: Phase plane plot for a four-impulse input. The system's resonant frequency is 10% higher than the anticipated natural frequency. The systems moves from the origin up to 1; around to 2; up to 3; around to 4; up to 5; around to 6; and back to (very near) the origin

4.5.5 Transfer Function Perspective

From the point of view of transfer function analysis, it is important to note that the impulse sequences are designed considering only the poles of the system. The zeros of the transfer function are not used for the implementation. Because the zeros do not need to be determined, the implementation of this technique is greatly simplified. Therefore, the technique will work on non-minimum phase systems, which are common when considering flexible machines.

Section 4.5.3 demonstrated that the impulse sequences worked by matching notches to the system resonances. Since numerator dynamics (zeros) do not cause resonant peaks, they can be ignored in designing a shaping sequence. Another way to show that numerator dynamics do not need to be included is to consider the transfer function of a linear system. Terms in the numerator indicate that the overall response is a

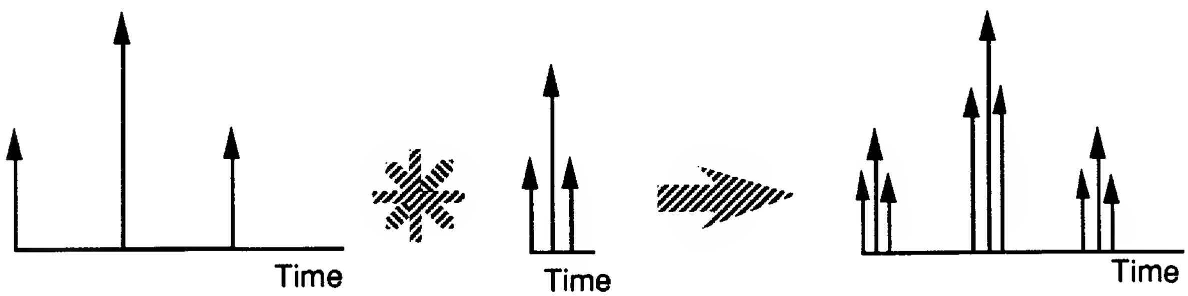


Figure 4.23: Vibration reduction for several modes. An example of convolving two three-impulse sequences together to form a single sequence that reduces vibration in two separate modes.

combination of the characteristic response plus derivatives of the characteristic response. Since the characteristic response is a decaying sinusoid (equation 3.1), the derivatives are also sinusoidal at the same frequency. If a shaping technique causes the characteristic response not to vibrate, then derivatives of the response must also not vibrate.

4.6 Including Higher Modes

The previous sections have assumed only one vibrational mode present in the system. However, the impulse sequence can easily be generalized to handle higher modes. If an impulse or pulse sequence is designed for each of the first two modes of a system independently, they can be convolved to form a sequence which moves a two-mode system without vibration. Figure 4.23 graphically shows this convolution for an example system.

The length of the resulting sequence is the sum of the lengths of the individual sequences. The sum, however, is an upper bound on the length of the two-mode sequence which can be generated directly. The direct solution is performed by simultaneously solving together the same equations that generated the two individual sequences with the addition of the extra terms. For example, if the four equations used to generate the

sequence in figure 4.4:

$$\begin{aligned}
\sum_{j=1}^{N} A_j e^{-\zeta\omega(t_{end}-t_j)} \sin\left(t_j\omega\sqrt{1-\zeta^2}\right) &= 0 \\
\sum_{j=1}^{N} A_j e^{-\zeta\omega(t_{end}-t_j)} \cos\left(t_j\omega\sqrt{1-\zeta^2}\right) &= 0 \\
\sum_{j=1}^{N} A_j(t_j) e^{-\zeta\omega(t_{end}-t_j)} \sin\left(t_j\omega\sqrt{1-\zeta^2}\right) &= 0 \\
\sum_{j=1}^{N} A_j(t_j) e^{-\zeta\omega(t_{end}-t_j)} \cos\left(t_j\omega\sqrt{1-\zeta^2}\right) &= 0
\end{aligned}$$

were repeated for a different frequency, a system of eight equations would result and could be solved for four unknown impulse amplitudes and times (plus the first, arbitrary impulse), yielding a five-impulse sequence. Note that originally only 3 vectors were needed so $N = 3$; when both sets of equations are used $N = 5$ so each of the equations gain some terms. The resulting sequence has four less impulses than the result of convolving the two independent sequences, and is always shorter in time. An arbitrary number of such sequences can be combined (either by convolution or by direct solution) to generate an input that will not cause vibration in any of the modes that have been included in the derivation.

4.7 Digital Implementation

The derivation presented above assumed that the timing of the impulses (the times at which the requested input is repeated into the system) could be specified exactly. If the system is digital, the spacing of the impulses is at fixed intervals — multiples of the sampling rate. Figure 4.24 demonstrates this problem assuming that a three impulse input is used. The middle impulse falls directly in between two sampling intervals. This causes a timing error. This section evaluates how well this technique fares when the sampling induced error δ becomes large.

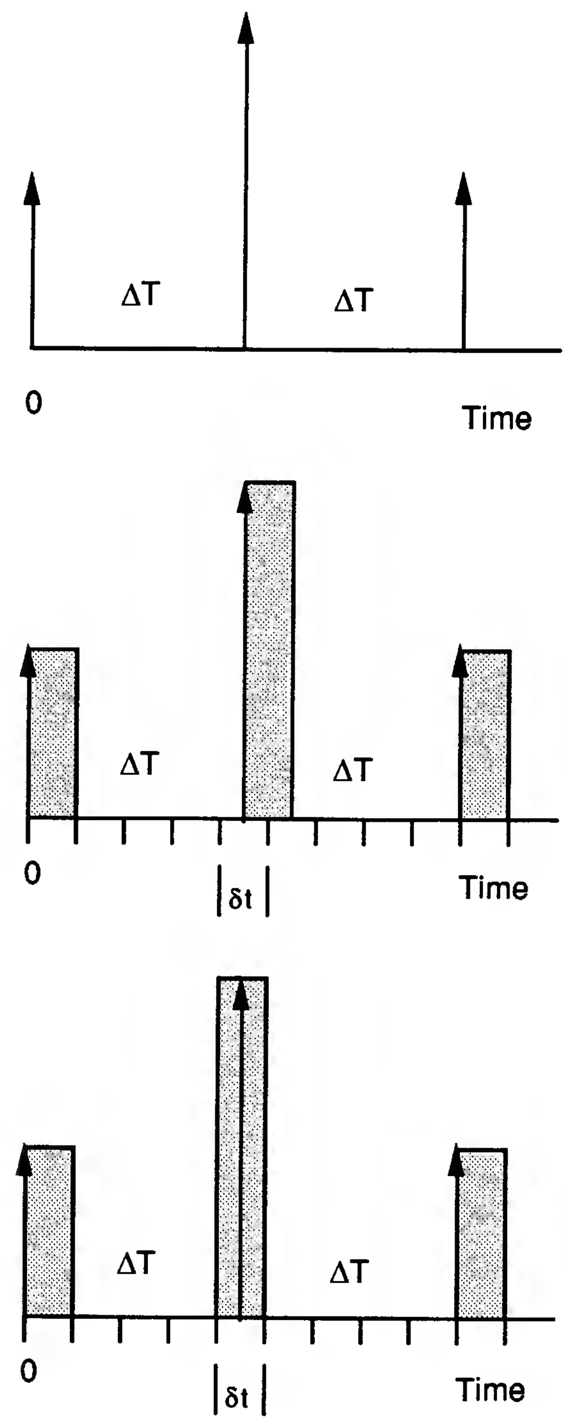


Figure 4.24: The problem of shaping inputs to digital systems. Top is the desired sequence. Middle: The digital timing of the system requires that the impulses do not all line up with the sampling intervals. Bottom: If the closest digital approximation is used (rounding to the nearest sampling interval), the impulse sequence is essentially translated as shown.

4.7.1 Quantizing the Error

The error will be measured as the vibration amplitude expressed as a percentage of the input magnitude (the same measure used in the robustness plots shown in section 4.3). The assumed impulse sequence has not been normalized, therefore, the three impulses do not sum to unity (this will be corrected later). Using the vibration amplitude expression from equation 3.3 for this situation shown in the bottom of figure 4.24 yields:

$$A_{\text{amp}} = \sqrt{\begin{array}{l} \left[1 + 2\cos\left(\omega\Delta T - \frac{\delta}{2}\right) + \cos\omega\Delta T\right]^2 \\ + \left[1 + 2\sin\left(\omega\Delta - \frac{\delta}{2}\right) + \sin 2\omega\Delta T\right]^2 \end{array}} \tag{4.6}$$

where ΔT is the half period of the oscillation, and δ is half of the sampling period.

Since $\omega\Delta T$ is equal to π, this expression becomes:

$$\begin{aligned} A_{amp} &= \sqrt{\left[2 + 2\cos\left(\pi - \frac{\omega\delta}{2}\right)\right]^2 + \left[2\sin\left(\pi - \frac{\omega\delta}{2}\right)\right]^2} \\ &= \sqrt{\begin{array}{l} \left(2 + 2\cos\pi\cos\frac{\omega\delta}{2} + 2\sin\pi\sin\frac{\omega\delta}{2}\right)^2 \\ + \left(2\sin\pi\cos\frac{\omega\delta}{2} - 2\cos\pi\sin\frac{\omega\delta}{2}\right)^2 \end{array}} \\ &= \sqrt{8 - 8\cos\frac{\omega\delta}{2}} \\ &= 4\sqrt{\frac{1 - \cos\frac{\omega\delta}{2}}{2}} \\ &= 4\sin\frac{\omega\delta}{4} \\ &\approx \frac{\delta}{\Delta T} \end{aligned} \tag{4.7}$$

This value for the residual vibration magnitude was derived for an impulse sequence that sums to 4 because the sequence was not normalized. In order to express the vibration error as a fraction of the input magnitude this factor of 4 must be divided out. The final estimate for the residual vibration error due to discretization is:

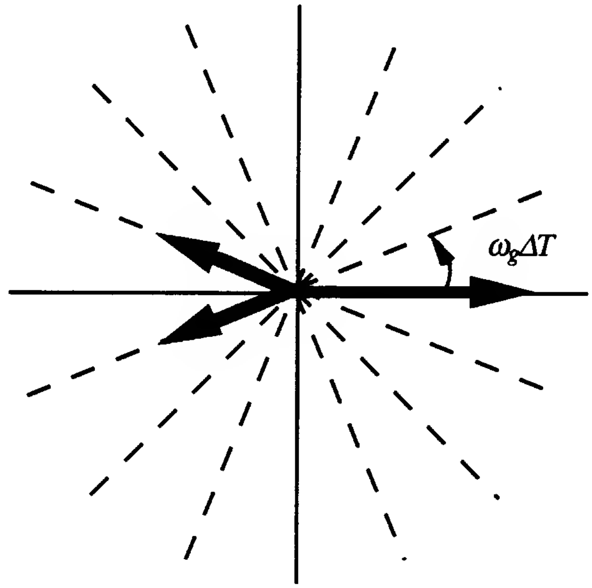


Figure 4.25: Vector diagram for a digital system. The vectors must be on one of the dotted lines. These correspond to the integral multiples of the sampling interval.

$$\text{Error} \approx \frac{\delta}{4\Delta T}$$

If this fraction is small for a particular digital system, then the digitization of the system can be ignored, and the impulses can be moved to the nearest sampling interval without inducing a significant vibration penalty. Small values for the error are typically less than 5% - 10%.

Sequences for Digital Systems

Once it has been determined that the error due to digitization is unacceptably large, a new form of the input sequence must be generated. This input sequence is made up of pulses which occur at integral multiples of the sampling interval. Figure 4.25 shows the effect of this added constraint on the location of the input pulses. The vectors must lie on the gray lines (at integral multiples of the sampling time).

It is desirable to put vectors in between these gray lines. Using the principle of

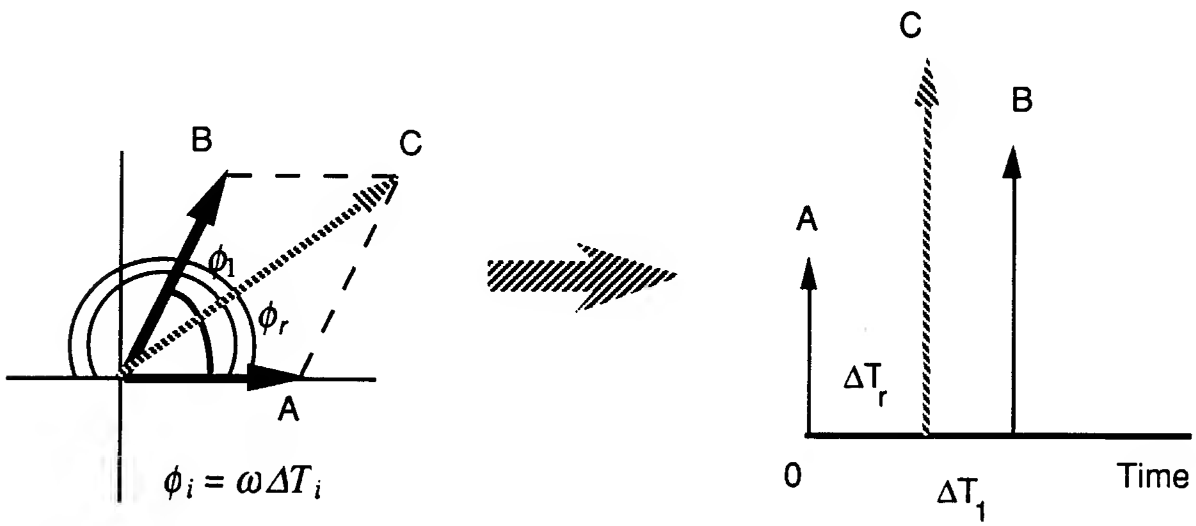


Figure 4.26: Adding impulses to create resultant impulses at the correct time. The two solid impulses add to form the hashed impulse. The hashed vector can replace the two solid vectors in the right-hand plot of the impulse sequence.

vector addition, it is possible to place two vectors (or more, if desired) on the gray lines such that they add to form the desired resultant which lies in between the gray lines. Figure 4.26 demonstrates the addition of two impulses to form the equivalent impulse between the two vectors.

Therefore, the solution for accommodating digital systems is to add at least one extra vector (one extra pulse to the sequence) so that the vibrations cancel. Figure 3.2 shows a sequence that was generated for a digital system with .08 second sampling interval. The sequence has the same robustness as the three-impulse sequence shown in figure 4.4. This five pulse sequence behaves identically to the three impulse sequences generated earlier. Additional pulses adjust for the timing constraints — **the same robustness and vibration reducing constraints have been met**. Because there are more pulses than constraint equations, the minimization routine additionally was able to minimize the second derivative constraints, thus yielding a **slightly** more robust sequence. The benefit of adding this additional constraint was minimal, however, it was retained since it has no "cost" associated with it. This demonstrates the relationship

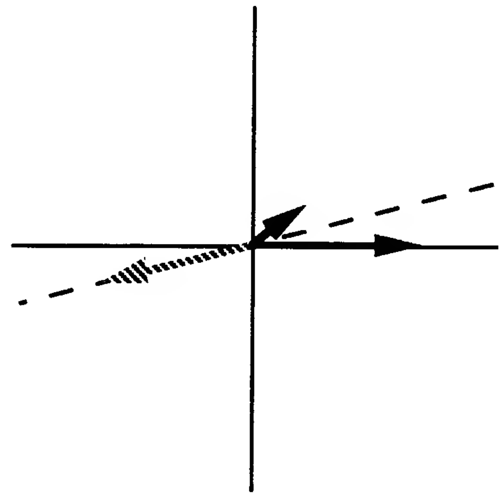


Figure 4.27: The effect of a negative impulse. The shaded vector is created with a negative impulse at the time corresponding to the dotted line in the upper-right quadrant. Note that because this vector flips over in to the lower left hand quadrant, it can cancel both the x and y components of the other vectors.

between the continuous, impulse sequences presented in this chapter and the digital, pulse formulation of chapter 3.

4.8 Negative pulses

In the previous sections, the impulse sequences consisted only of positive impulses. If this constraint is relaxed, shorter sequences result. As will be demonstrated, for a given robustness, a sequence can be made arbitrarily short. The price that is paid for the shorter sequences (and, therefore, faster system response) is actuator effort. The limiting factor, therefore, in shortening the sequences is saturation of the system actuators.

An examination of equations 3.6 will reveal that an infinity of shorter sequences exists and that the sequences can be arbitrarily short. Figure 4.27 graphically demonstrates this result on a vector diagram. The shaded vector is created by a negative impulse along the dotted line in the upper-right hand quadrant. Since the magnitude is negative, it has the same functional effect as placing a positive vector where the shaded vector is.

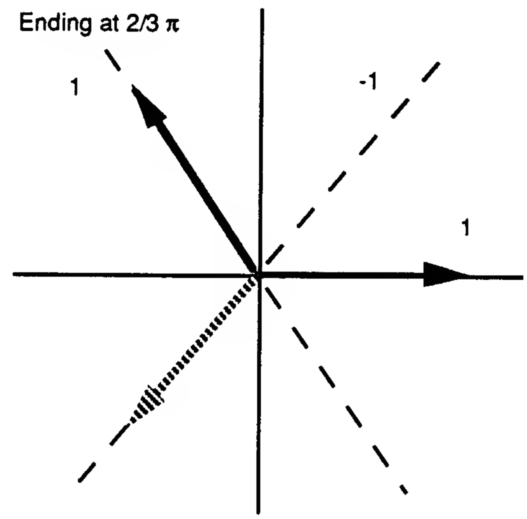


Figure 4.28: Shortening a sequence with negative impulses. This is a vector diagram of a sequence that has been shortened by 33%. The second impulse is negative, therefore, it points into the third quadrant. This enable it to vector-cancel with the other vectors. This sequence has the same robustness qualities as the positive, **two-impulse** sequences.

Since each system is different, no one short sequence can be used. A sequence must be shortened until actuator limits are reached or derived directly by including additional constraints that are imposed by the system. However, in this section, a two-impulse sequence that is shorter than the one shown in figure 4.2 is derived.

Figure 4.28 shows this new sequence that is equivalent (in terms of robustness) to the two-impulse sequence. This sequence does not exceed any control limits because when it is convolved with a step, it never exceeds unity. This sequence is 33% shorter than the two-impulse sequence derived above yet it is functionally equivalent. Note that the magnitude of the impulses (assuming no damping) is one. The equations that were solved to yield this sequence (equations 3.6) have no solutions that result in a shorter sequence without the impulses exceeding unity magnitude (without damping — the sequence can be made slightly shorter as a function of damping). Therefore, the sequence of figure 4.28 is the shortest sequence that is equivalent to the two impulse input that does not exceed control limits.

Figure 4.29 shows a shortened equivalent to the three-impulse sequence derived above. The sequence shown in the figure is for a digital system with .08 second sampling time and a .6 hz natural frequency. This sequence was generated using a minimization routine that shortened the sequence length while imposing the constraints used to generate the three-impulse sequence. The cost function was the second derivative expression so that the maximum robustness was obtained. Some additional constraints were included in order to require that the system not saturate during a unit step input. Note that this one sequence is 10% shorter than the equivalent three-impulse sequence. This sequence demonstrates the combination of several of the variations presented in this chapter for non-standard systems.

For many applications, there is no requirement to shorten the sequences, therefore, the sequences that only have positive impulses will often be preferred.

4.9 Multiple Joint Actuation

4.9.1 Linear Systems

One important question that must be addressed in using this technique is the effect of simultaneously shaping two separate joints of a machine. The technique would be of limited utility if it could only be used on single joint machines. This last section will discuss the effect of shaping several machine inputs simultaneously.

First, a generalized linear system is considered. Figure 4.30 shows a three-mass, two-input example system. The output that will be considered is x_3. Since any vibration of M_3 must appear in the other coordinates as well (according to Newton's Law), it makes no difference which output is examined. The transfer functions for the input-output relationships at x_3 are:

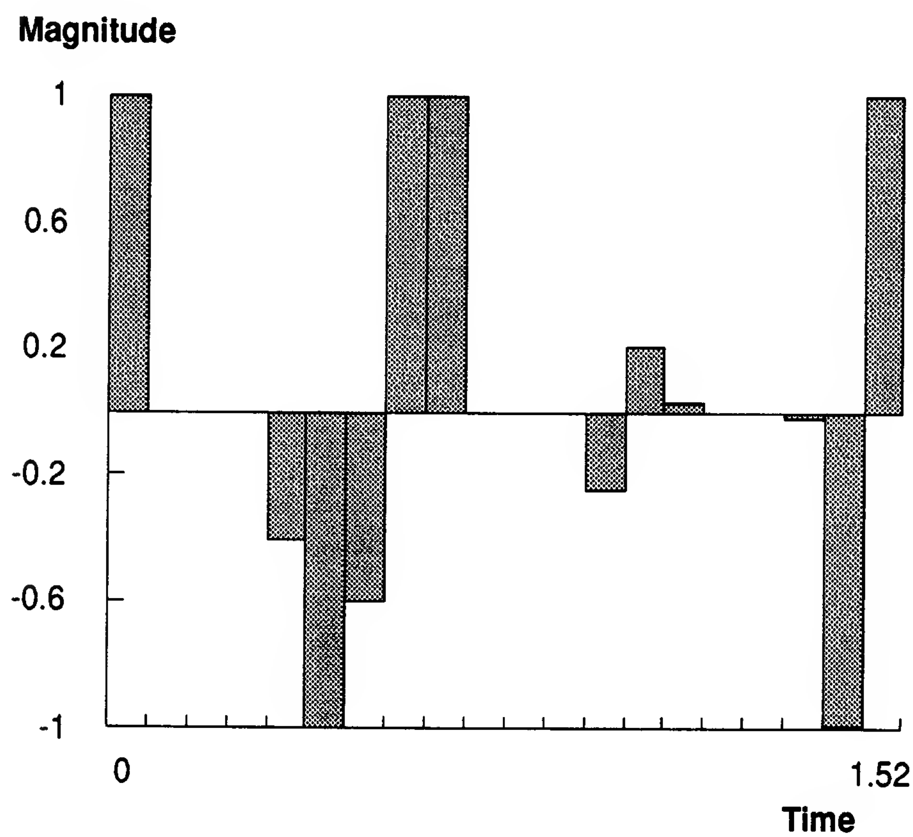


Figure 4.29: Negative sequence for a digital system. This figure demonstrates the combination of several techniques. The sequence is shortened by allowing the impulses to be negative. The system is digital, therefore, additional pulses must be added to satisfy all of the constraints. The system natural frequency is .5 hz; the sampling time of the system is .08 seconds.

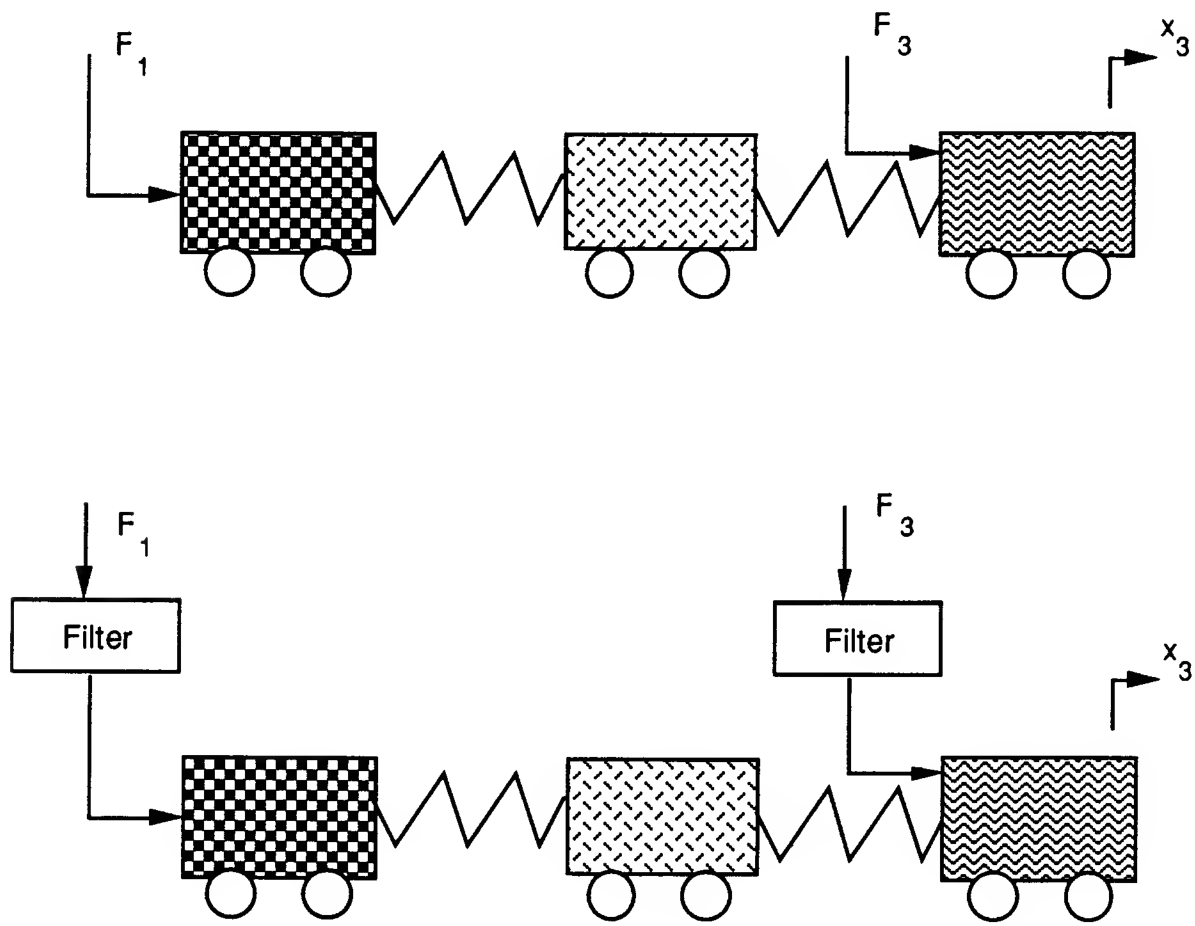


Figure 4.30: Multiple actuation on a linear system. The same filter is used for both inputs to the system because the characteristic equation is the same.

$$\frac{x_3}{F_3} = \frac{\text{Numerator}_3}{\text{Characteristic}} \tag{4.8}$$

$$\frac{x_3}{F_1} = \frac{\text{Numerator}_1}{\text{Characteristic}} \tag{4.9}$$

Note that the characteristic equations are the same. This fact is a characteristic of all linear systems. Since the shaping is designed for the resonances of the system (the natural frequency and damping ratio), the characteristic equation fully determines the shaping sequence that will be used. Because the two transfer functions have the exact same characteristic equations, the prefilters on each of the inputs are identical.

Using an informal inductive explanation, the prefiltering on one input can be shown not to effect or interfere with prefiltering on the other inputs. The argument is as follows:

- A prefilter on input F_1 will cause the output x_3 to change without vibration.
- The same prefilter on input F_3 will cause the output x_3 to change without vibration.
- All inputs are superposable. The output at x_3 due to both an input at F_1 and an input at F_3 is the sum of the outputs due to inputs F_1 and F_3 put into the system alone. The response caused by one input **cannot** effect the other input. This would be a violation of superposition and linearity.

Therefore, for linear and quasi-linear systems, any number of inputs may be prefiltered using the same impulse sequence to shape the inputs. The next section will show how multiple joint actuation applies to geometrically nonlinear systems. Appendix D discusses additional nonlinear system applications.

4.9.2 Vibrationless Cartesian Motion from Non-Cartesian Machines

When the system that is to be controlled is a cartesian machine, the techniques of the previous Chapters apply. However, often the system is not cartesian and, therefore, straight line motion must be achieved by computation of joint trajectories for cartesian motion. The problem that will be addressed in this section is the effect of shaping on the overall endpoint trajectory of the machine. The sample system that is used for the results in this section is the two-link manipulator shown in figure 4.31. This system has rigid links and lumped parameter springs at the joints. The parameters for this model used in this section are:

- length of link 1 and 2: 1.0
- mass of link 1 and 2: 1.0

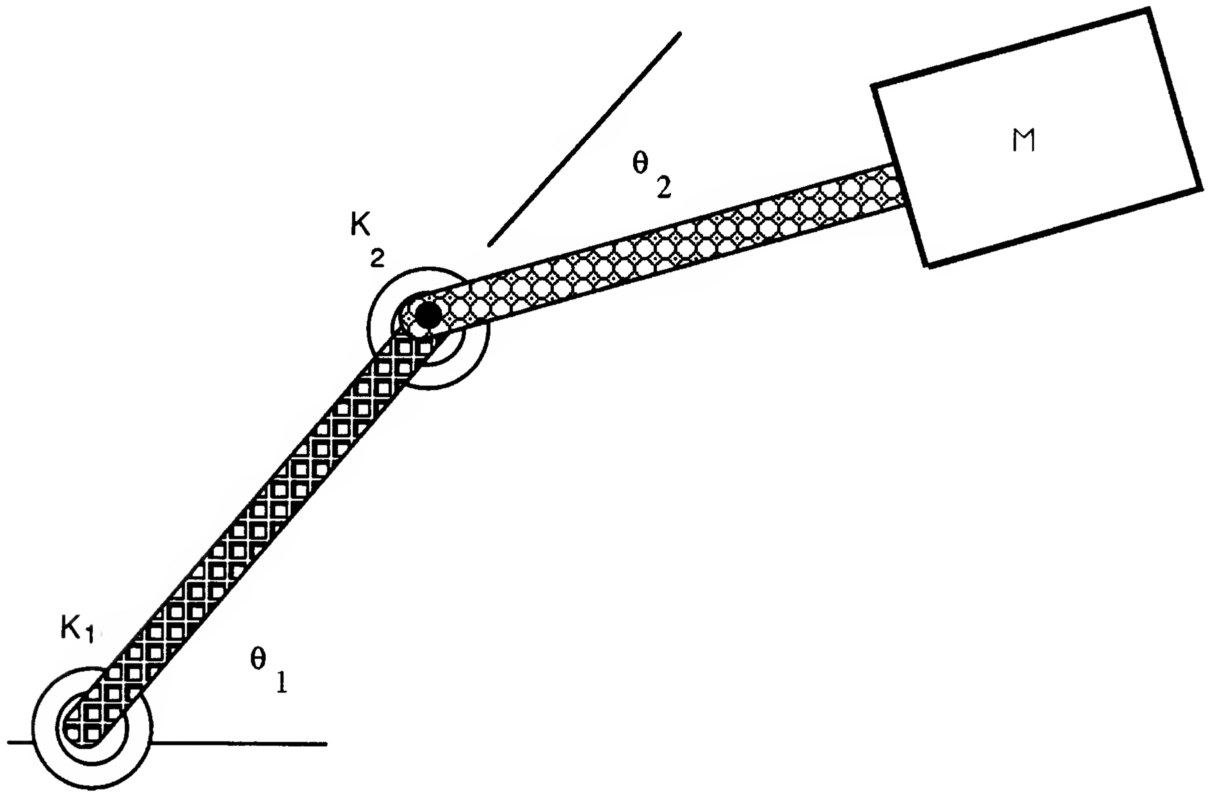


Figure 4.31: Nonlinear test model used to evaluate various techniques.

- inertia of link 1 and 2: 1.0
- end mass: 12.0
- end mass inertia: 9.0
- spring 1: 50.0
- spring 2: 100.0

Damping was excluded so that the effect of shaping is isolated from the effects of damping.

Two main approaches will be examined. The first is to determine the straight line joint trajectories that would be required assuming that the signals were not to be shaped. Next, these joint trajectories are shaped so that they become vibration-reducing. The advantage of this approach is that the vibration control is the best possible (keeping all

other factors constant) . The disadvantage is that the trajectories are not **theoretically** exact straight line trajectories. However, the original "straight-line" trajectories are not perfectly straight either [108]. Intermediate points are computed on a straight-line trajectory and non-straight, joint-interpolated motion is used between these points. Therefore, cartesian trajectories in practice are only as straight as the available computation allows (Paul [108] states that joint-interpolated motion requires roughly 1% of the computation of cartesian motion). Shaping the trajectory does not significantly alter the cartesian nature of the input, especially as more intermediate points are used. Additionally, since the shaped trajectory does not have the unwanted vibration in the output, the **actual** endpoint position will be far closer to "straight" than the unshaped trajectory. (If the vibration was not causing problems, shaping would never have been considered for that system!) Figure 4.32 shows a schematic of this arrangement.

The second approach is to shape the cartesian trajectories and then convert them to joint trajectories. This approach guarantees that the trajectories will be as straight as possible (keeping all other factors constant). The drawback of this approach is that the vibration reduction is slightly degraded. Figure 4.33 shows a schematic of this arrangement.

Under many conditions and for many systems, either approach is acceptable although not exact. Figure 4.35 shows a simulation of the two link system when the Jacobian calculation is performed before the vibration shaping. Figure 4.36 shows a simulation of the two link system when the Jacobian calculation is performed after the vibration shaping. Both modes of the system are included in the command shaping. The slightly longer, convolved version of the two mode formulation is used. Note that both configurations are extremely straight with little vibration. The system was assumed to be velocity limited and is therefore started using one sequence and stopped with a different sequence according to the technique described in appendix D.2. The next section shows similar results from cartesian motion on the space shuttle remote manipulator arm.

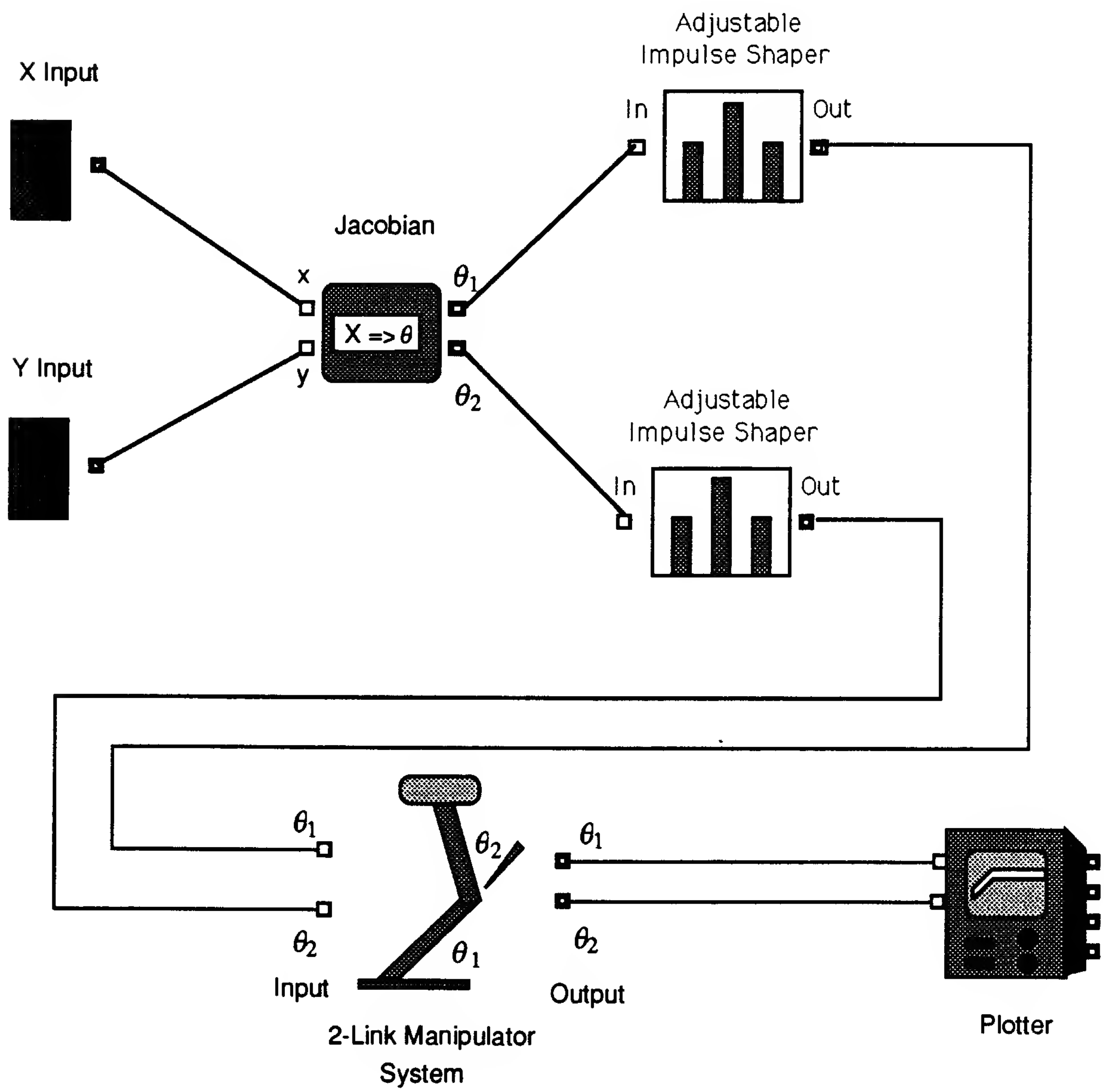


Figure 4.32: Cartesian Motion — shaping before Jacobian. This is the schematic of the simulations that were performed when the shaping algorithm was placed before the Jacobian calculation. The Jacobian calculation converts the cartesian command to joint angles.

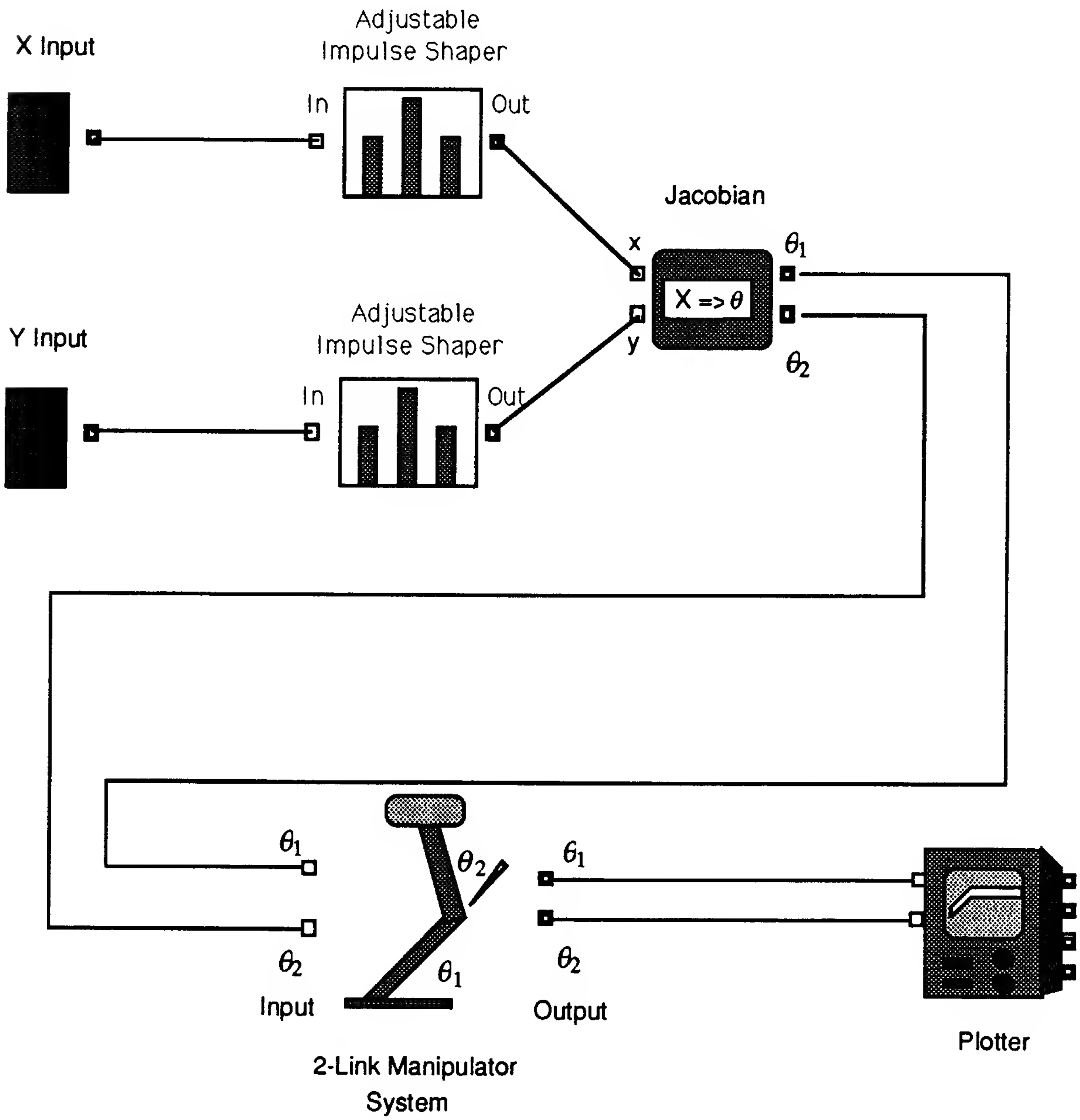


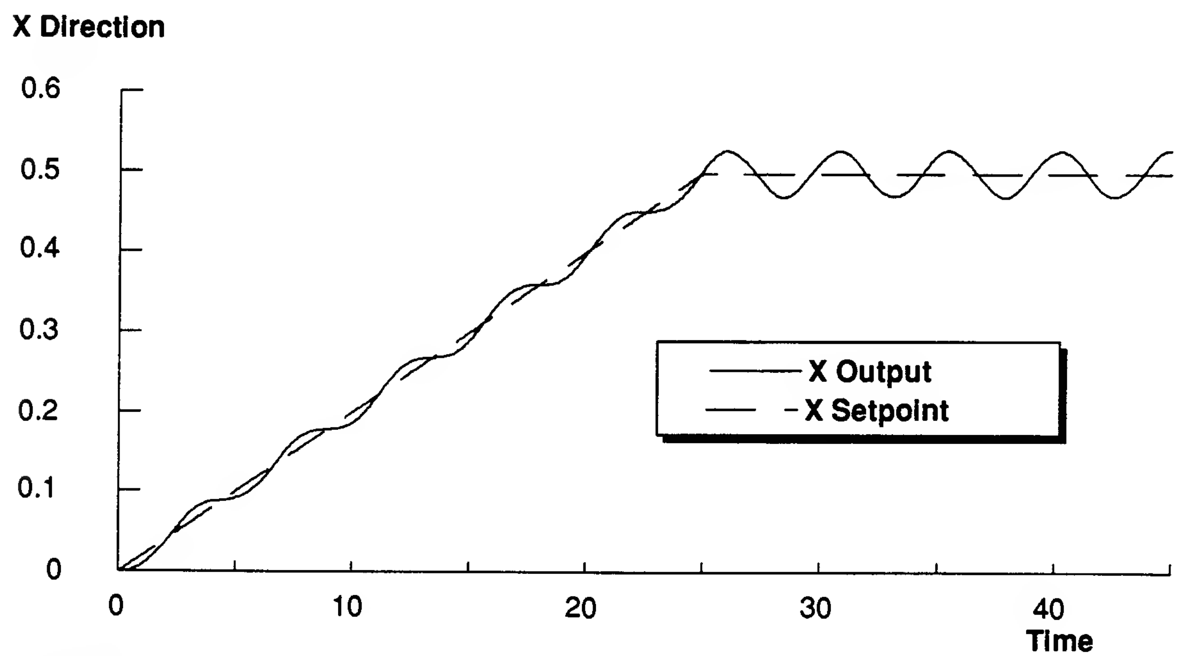
Figure 4.33: Cartesian Motion — shaping after Jacobian. This is the schematic of the simulations that were performed when the shaping algorithm was placed after the Jacobian calculation. The Jacobian calculation converts the cartesian command to joint angles.

The plots of figures 4.34, 4.35, and 4.36 show that adding shaping to this sample, undamped, geometrically-nonlinear system reduces the residual vibration by approximately a factor of fifty. Both configurations of Jacobian calculations had comparable (and small) residual vibration levels. When the Jacobian calculation is performed as the last step before sending the command to the system, the slewing motion is straight and follows the commanded y position (except during the acceleration and deceleration phase). When the Jacobian calculation is performed first, the slewing is straight but it does not follow the commanded y position (figure 4.36).

4.10 Cartesian Motion Using the Shuttle Arm.

Figure 4.37 shows a cartesian move on the space shuttle arm. The details of this and the other models used for theory verification are given in section 2.3. The joint trajectories are calculated first and then are shaped. The plot shows the motion in the y-direction. Figure 4.38 shows the x and z direction motion for the same cartesian move. The command is only in the y-direction. These plots demonstrate that even without the preshaping, the shuttle's "cartesian" motion is not straight, and the vibration amplitude effects the straightness of the motion to a much larger extent than the alteration of the joint trajectories from shaping.

Figure 4.39 shows a comparison of the energy consumed by the shuttle manipulator during the two moves. A 20% savings in energy was realized by not inducing vibration in the arm. This energy savings has significant implications for space systems like the shuttle and space station. Since energy in space is expensive (the shuttle, for example must carry its own fuel) the energy savings alone may justify shaping of the command input for the reduction of vibration.



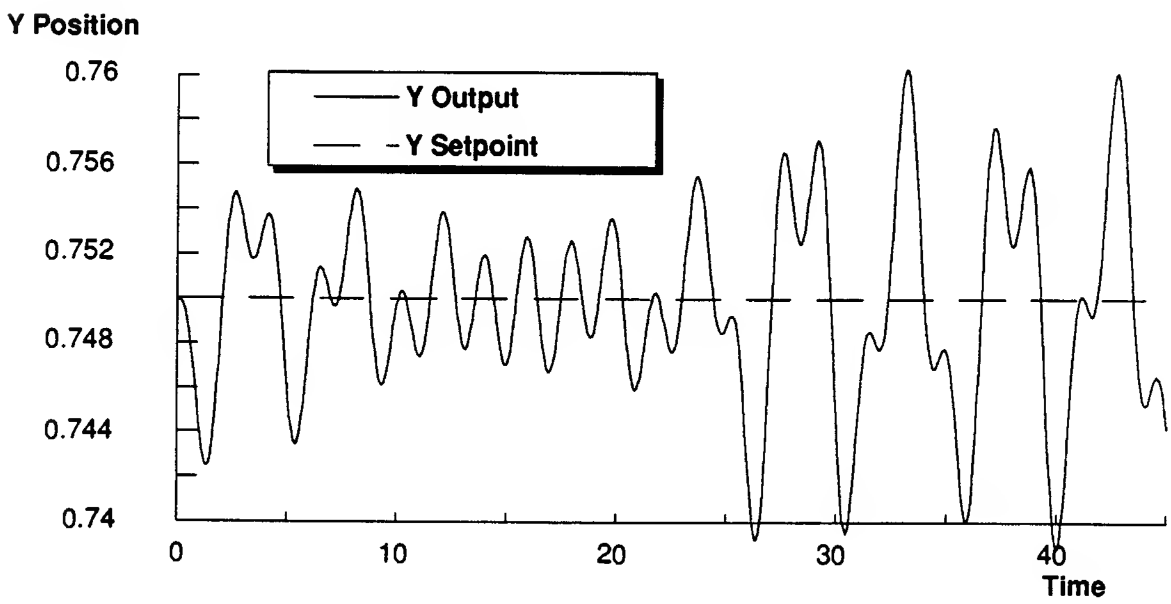


Figure 4.34: Cartesian Motion from a two link manipulator — unshaped results. The top plot shows the X motion of a system for which no shaping is performed. The bottom plot shows the Y motion of the same system. The system has no damping so that the vibration-reducing effects are isolated.

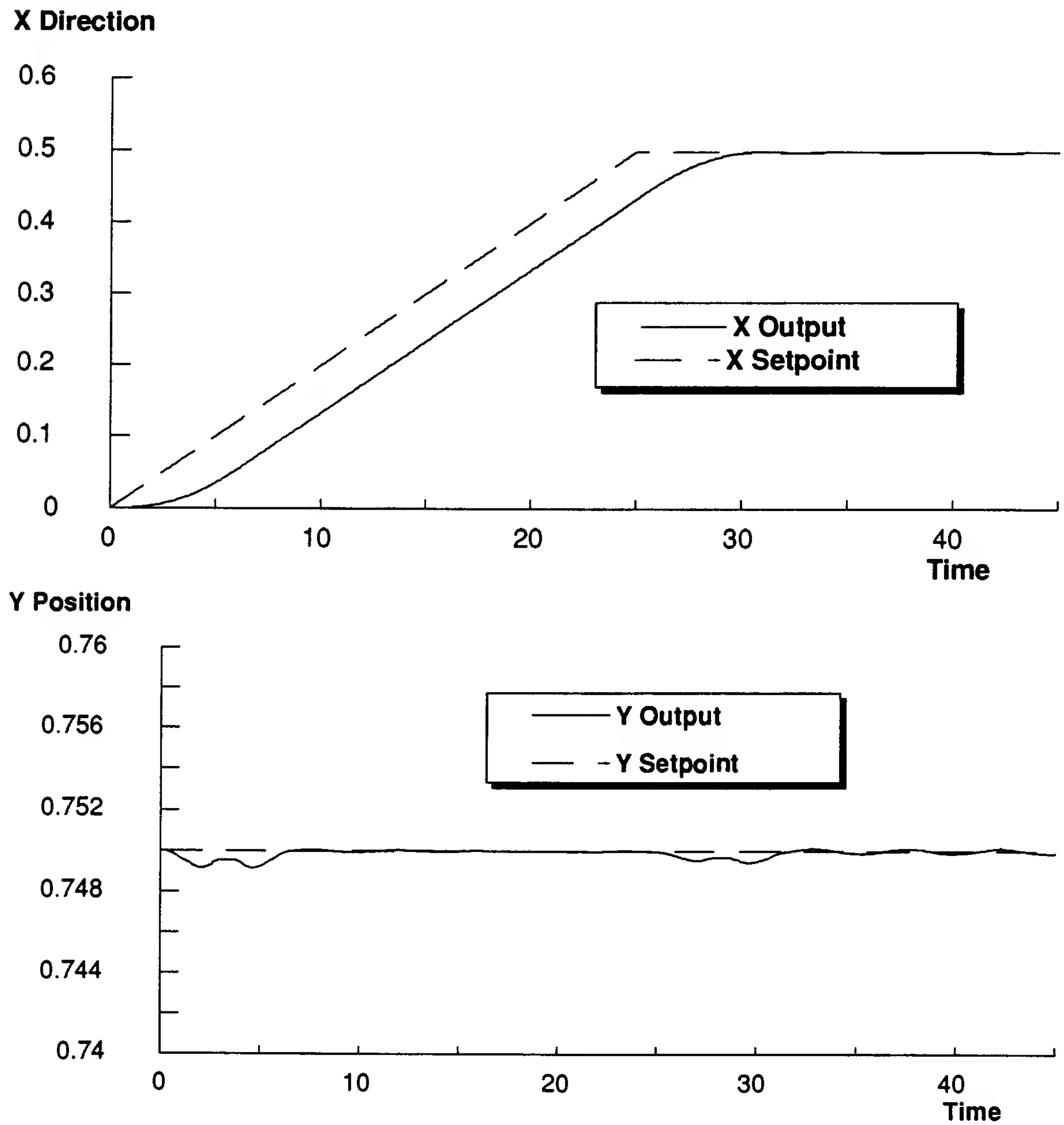


Figure 4.35: Cartesian Motion from a two link manipulator — shaping first. The top plot shows the X motion of the two-link system for which the shaping was performed **before** the Jacobian calculation (figure 4.33). The bottom plot shows the Y motion of the same system. The system has no damping so that the vibration-reducing effects are isolated.

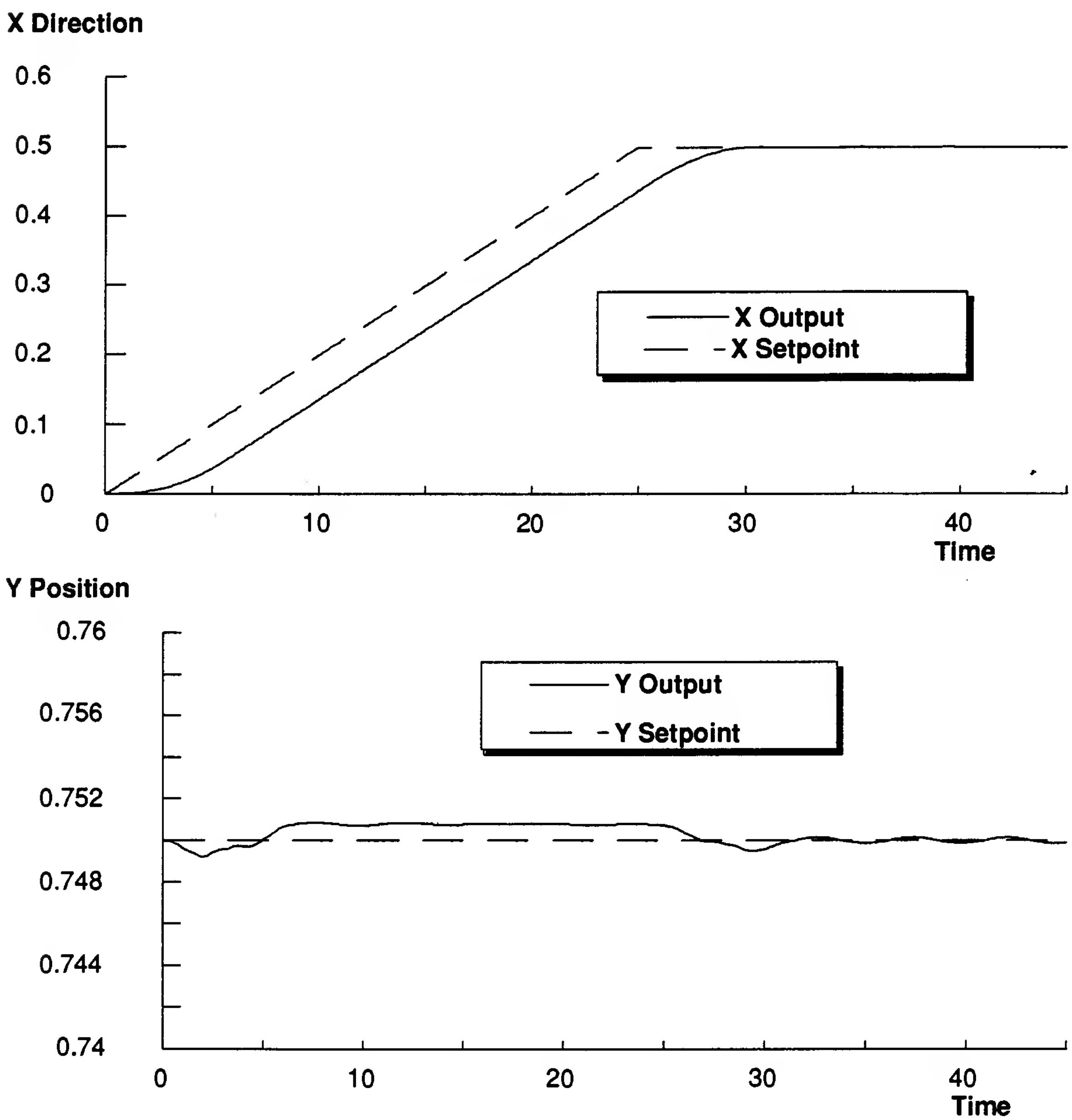


Figure 4.36: Cartesian Motion from a two link manipulator — shaping last. The top plot shows the X motion of the two-link system for which the shaping was performed **after** the Jacobian calculation (figure 4.32). The bottom plot shows the Y motion of the same system. The system has no damping so that the vibration-reducing effects are isolated.

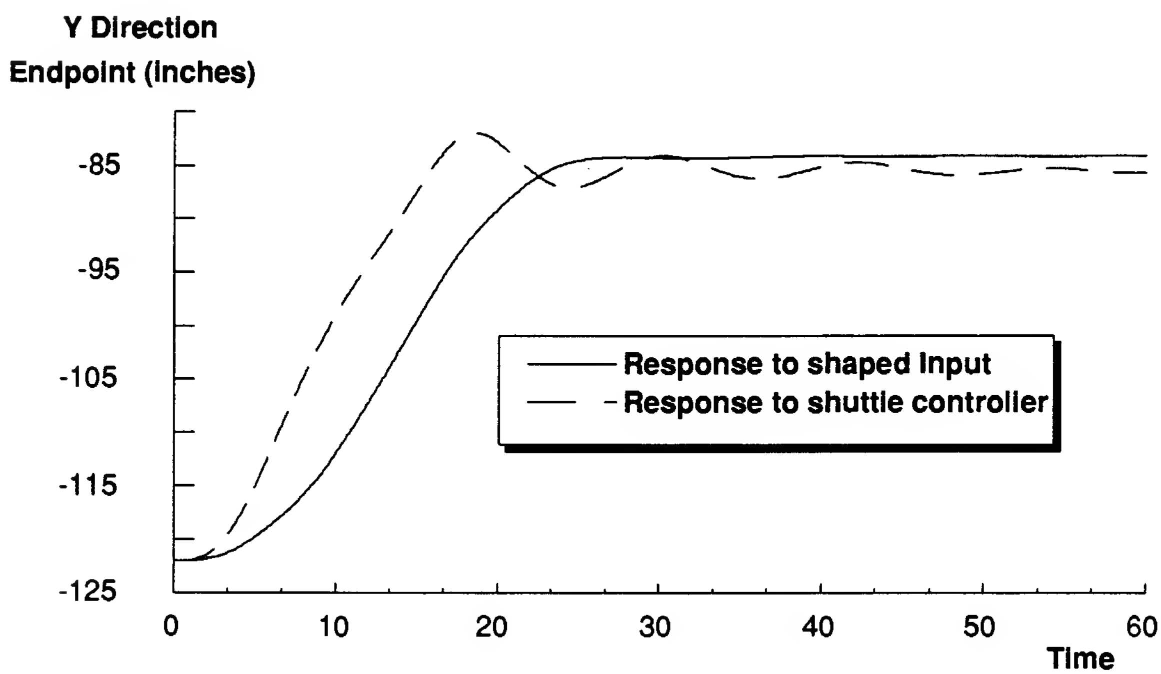


Figure 4.37: Cartesian motion of the shuttle manipulator. The command to the shuttle was a straight-line motion (step) in the y direction. The dashed line is the unaltered RMS controller. The solid line is a shaped input. The data is shown is motion in the y direction. The next figure shows the x and z motion during the same move.

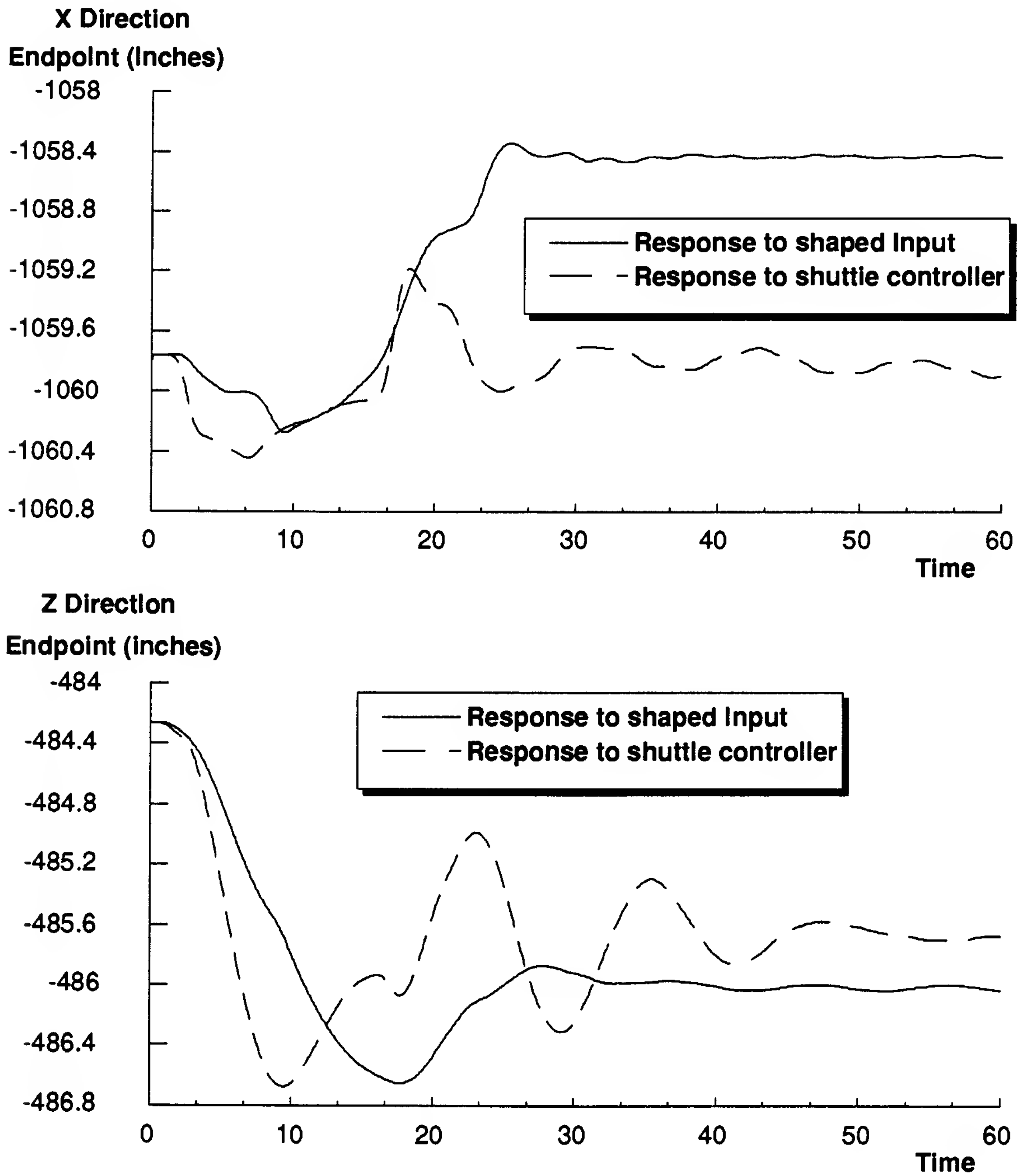


Figure 4.38: Cartesian motion of the shuttle manipulator. The command to the shuttle was a straight-line motion (step) in the y direction. The dashed line is the unaltered RMS controller. The solid line is a shaped input. The motion perpendicular to the commanded motion is shown. The previous figure shows the motion in the commanded direction. Note that neither move (with or without shaping) is extremely "straight" on the DRS.

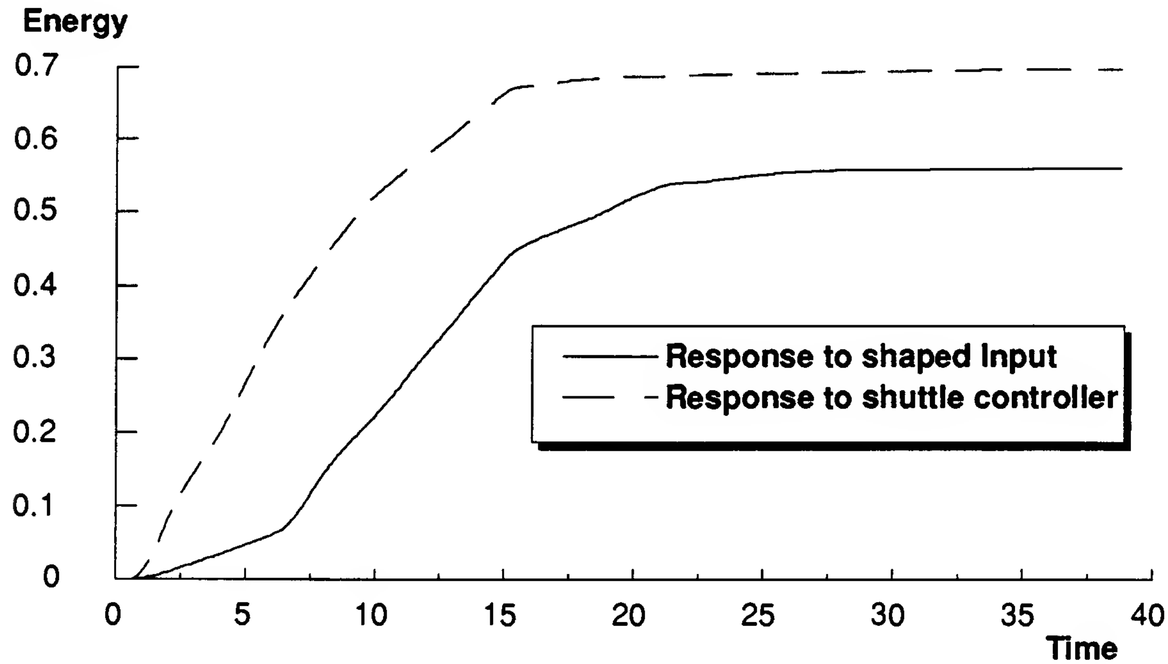


Figure 4.39: Energy plot for straight-line motion. This figure demonstrates the energy savings by using the shaping technique of chapter 4.

4.11 Trajectroy Alterations Caused by Shaping

This chapter has shown that shaping of an input trajectory can create a new trajectory that does not cause the system to vibrate. This section will examine how close the new trajectory is to the original, unshaped commanded trajectory; shaped and unshaped trajectories will be compared in order to reveal the effect of shaping. The system response will not be considered in this section. Chapter 5 will show that systems closely follow the altered trajectory without vibration.

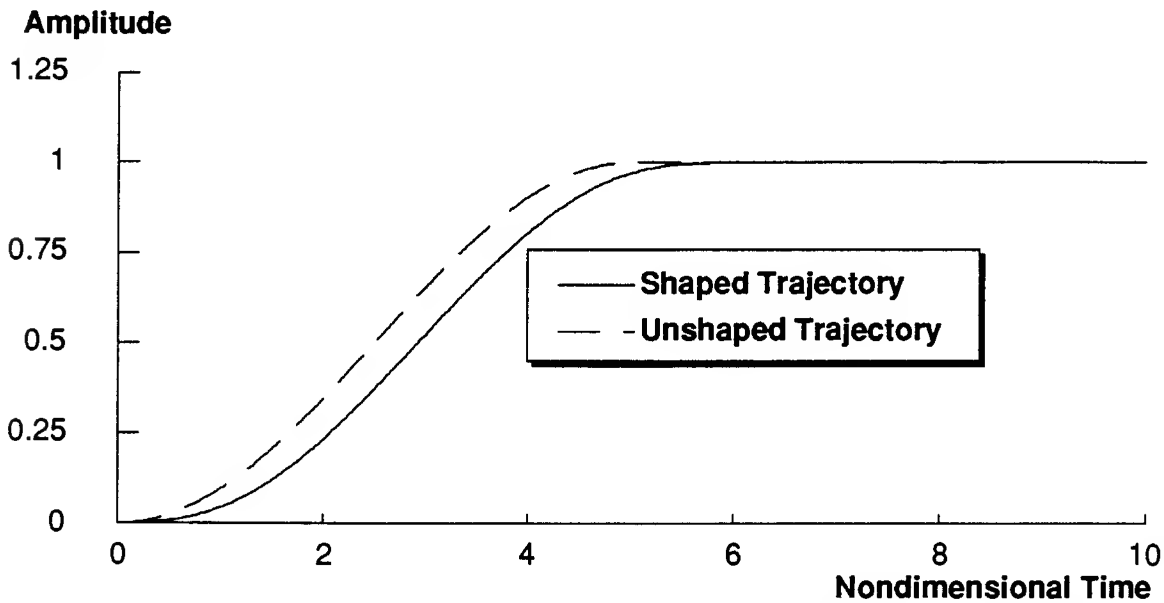
First, a single axis command is considered. Figures 4.40 and 4.41 compare several similar trajectories. A versine of amplitude from 0.0 to 1.0 units is generated. The shaping sequence is a three-impulse sequence designed for a system with a .1 damping ratio. The results are normalized so that the time of one system period is 1.0 on the normalized time axis. The rate at which the transition is made is varied from .1 periods

to 5 periods. Note that when the transition is extremely short compared to the period of the system, the shaped trajectory does not have the same basic shape as the command. This corresponds to the situation when the input trajectory has high frequency content compared to the system natural frequency. When the transition is slow (low frequency input trajectory), the shaped trajectory has the same basic shape as the requested input. The shaped trajectory lags behind the command by one period of the system. However, for all moves, the shaped trajectory ends at the same value as the command.

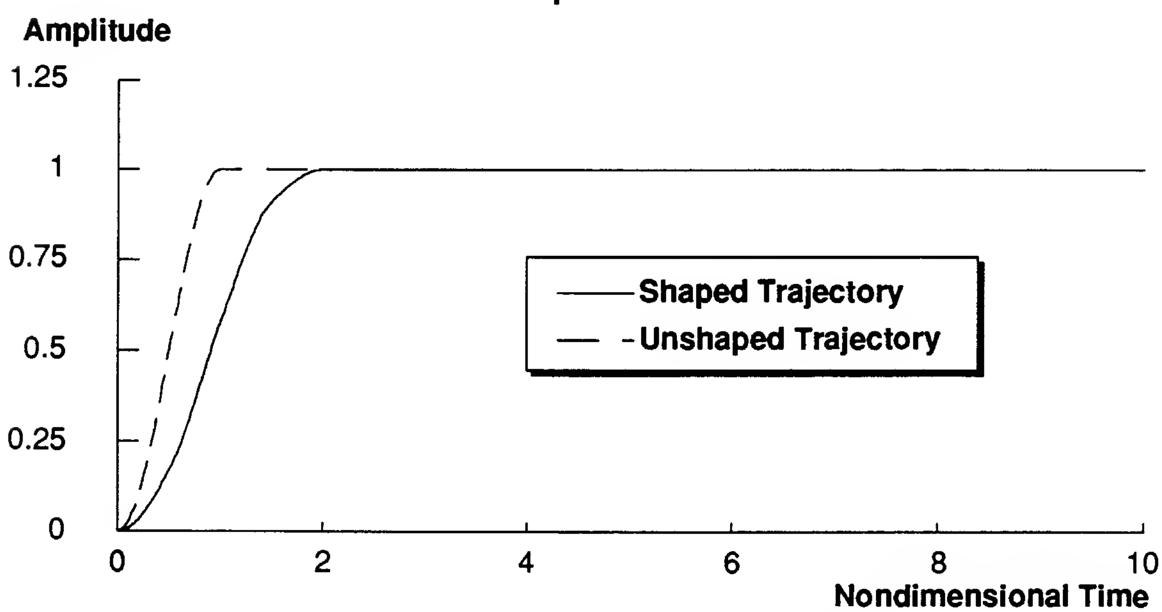
The next situation that is considered is the effect of shaping on cartesian trajectories. The system that is considered is the two-link, planer manipulator shown in figure 4.31 with one unit link lengths. A requested cartesian trajectory is generated. It is then converted to a joint trajectory. Next, the joint trajectories are shaped. The resulting, shaped joint trajectories are then converted back to a cartesian reference frame and compared to the original, requested command. Figure 4.42 shows a schematic of this test. Figure 4.43 through 4.46 summarize the results of this test. A series of versine x-coordinate commands were given to the system while the y-coordinate was commanded to remain at zero (a straight-line motion in the x-direction). The trajectory is nearly straight when the transition time is long compared to the system's period of vibration. When the transition is fast, the trajectory does not remain straight. Note that the figures are to scale so that the each curve shown is the motion of the endpoint in the workspace of the two-link manipulator.

The cartesian situation in which the trajectory is shaped prior to resolving the joint commands is not considered separately. The x-coordinate motion for this situation is identical (by definition) to that of the single-axis trajectory considered in figures 4.40 and 4.41. The y-coordinate is unaltered by shaping and remains .5 (for the example shown) for all time.

As the requested trajectory becomes fast relative to the period of the system, the trajectory following degrades. However, it is important to note that the ability of the



Versine Input 5.0 Period Duration



Versine Input 1.0 Period Duration

Figure 4.40: Single axis trajectory comparison. The unshaped trajectory is convolved with a three-impulse sequence designed for a sytem with a .1 damping ratio. One unit of nondimensional time corresponds to one period of oscillation used to design the three-impulse sequence. The top plot shows the result for a relatively slow versine input that transitions from 0 to 1 in 5 periods. The bottom plot shows the result for a relatively fast versine input that transitions from 0 to 1 in 1 periods.

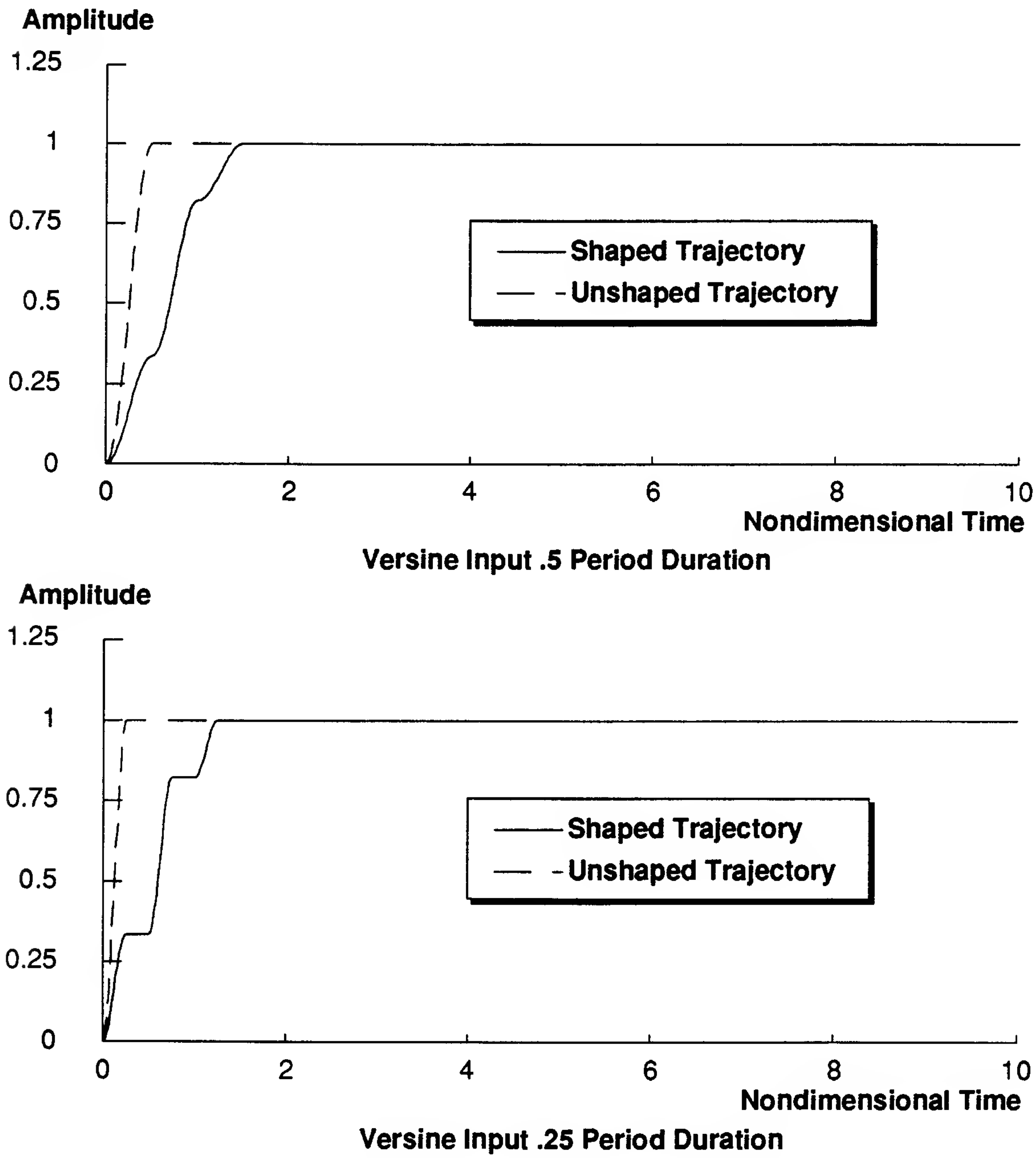


Figure 4.41: Single axis trajectory comparison. The unshaped trajectory is convolved with a three-impulse sequence designed for a sytem with a .1 damping ratio. One unit of nondimensional time corresponds to one period of oscillation used to design the three-impulse sequence. The top plot shows the result for a fast versine input that transitions from 0 to 1 in .5 periods. The bottom plot shows the result for an extremely fast versine input that transitions from 0 to 1 in .25 periods.

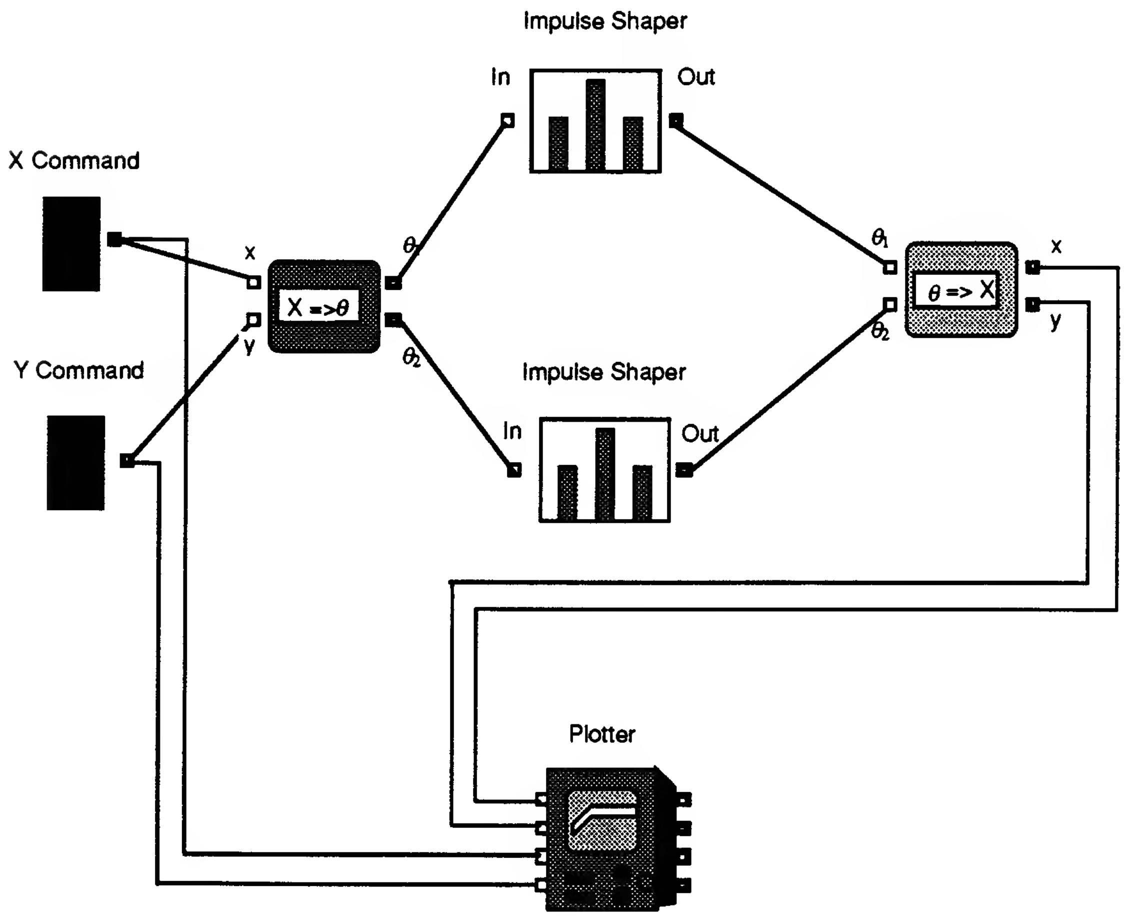


Figure 4.42: Schematic of the cartesian trajectory experiment. A variable-duration versine input is commanded in the x-direction while no motion is commanded in the y-direction. The joint trajectories are shaped and then converted back to cartesian corrdinates before being plotted.

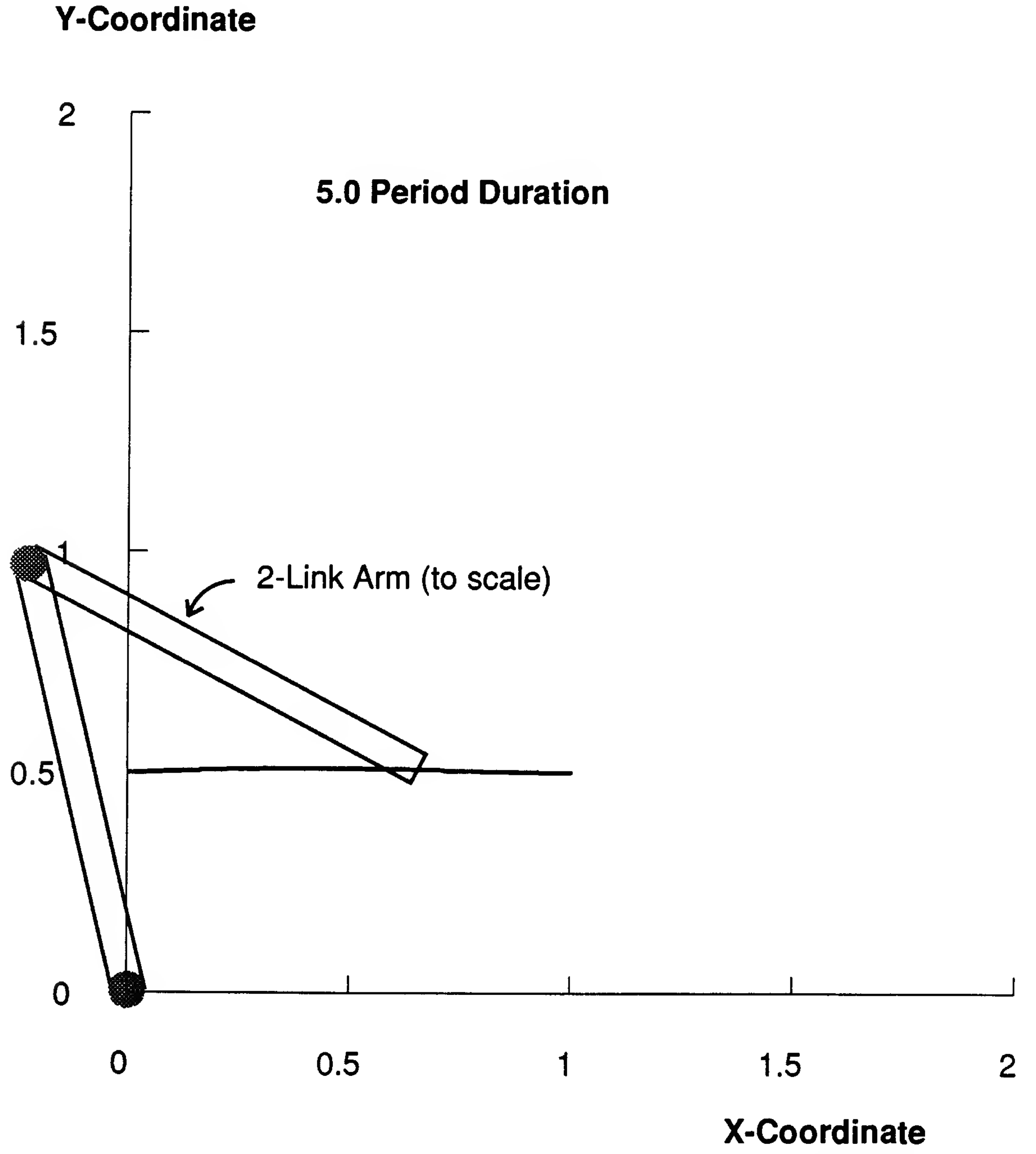


Figure 4.43: Cartesian trajectory comparison. The commanded trajectory is a fast versine in the X-direction with a constant .5 value commanded for the Y-direction. The The versine transitions from 0 to 1 in 5 periods. Note that this is not a time plot. The manipulator geometry is shown for a single point along the trajectory.

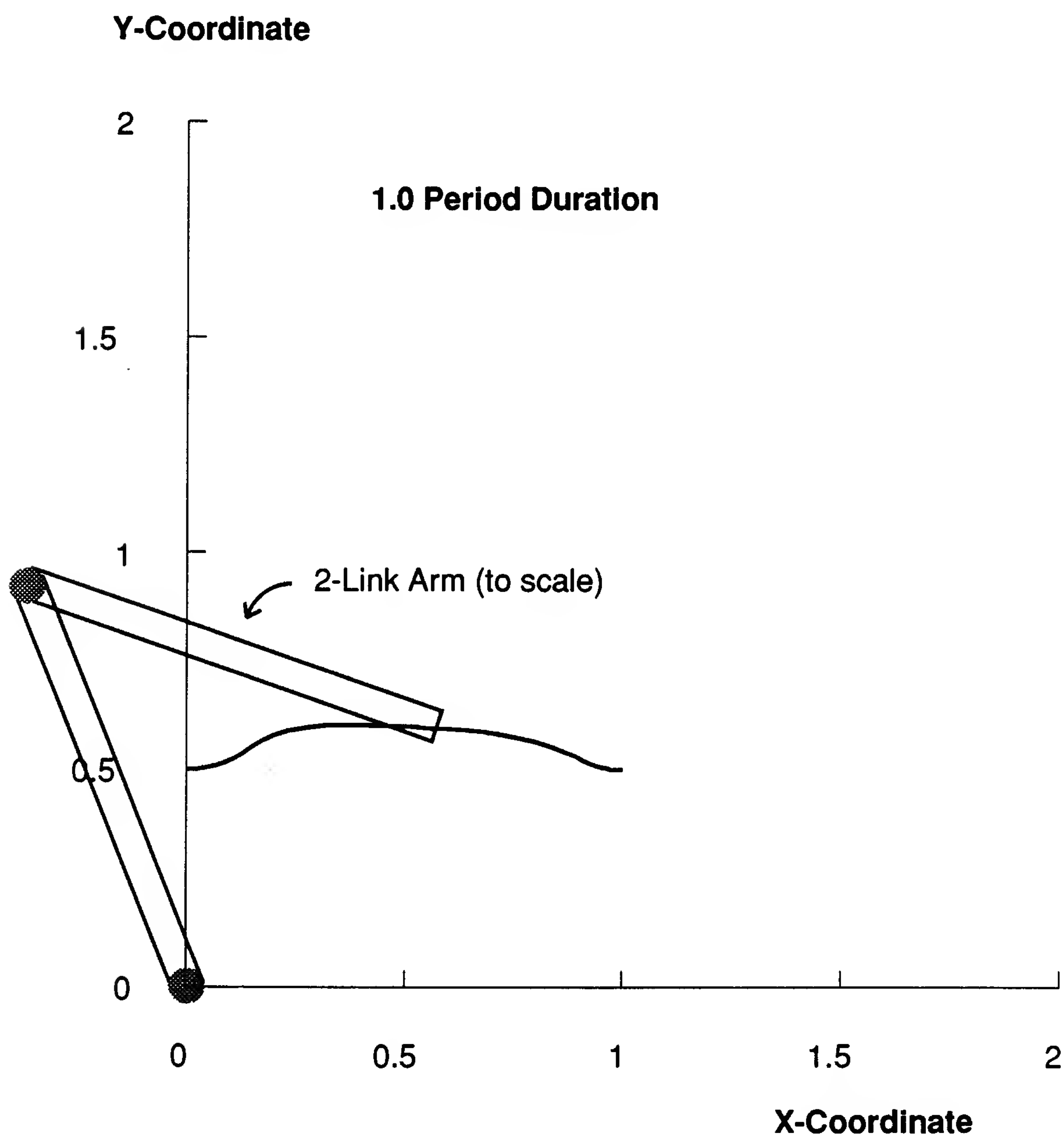


Figure 4.44: Cartesian trajectory comparison. The commanded trajectory is a fast versine in the X-direction with a constant .5 value commanded for the Y-direction. The The versine transitions from 0 to 1 in 1 period. Note that this is not a time plot. The manipulator geometry is shown for a single point along the trajectory.

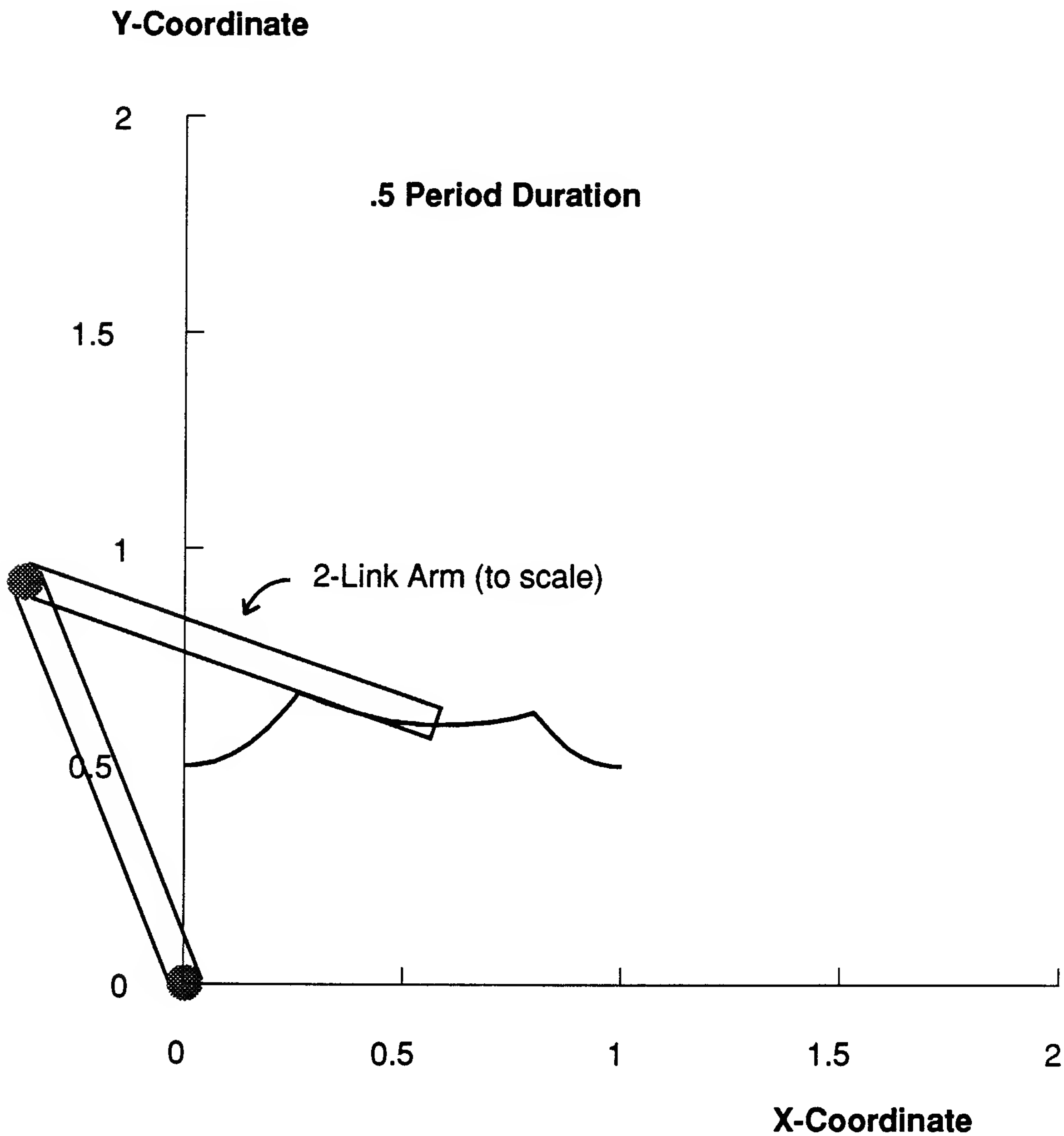


Figure 4.45: Cartesian trajectory comparison. The commanded trajectory is a fast versine in the X-direction with a constant .5 value commanded for the Y-direction. The The versine transitions from 0 to 1 in .5 periods. Note that this is not a time plot. The manipulator geometry is shown for a single point along the trajectory.

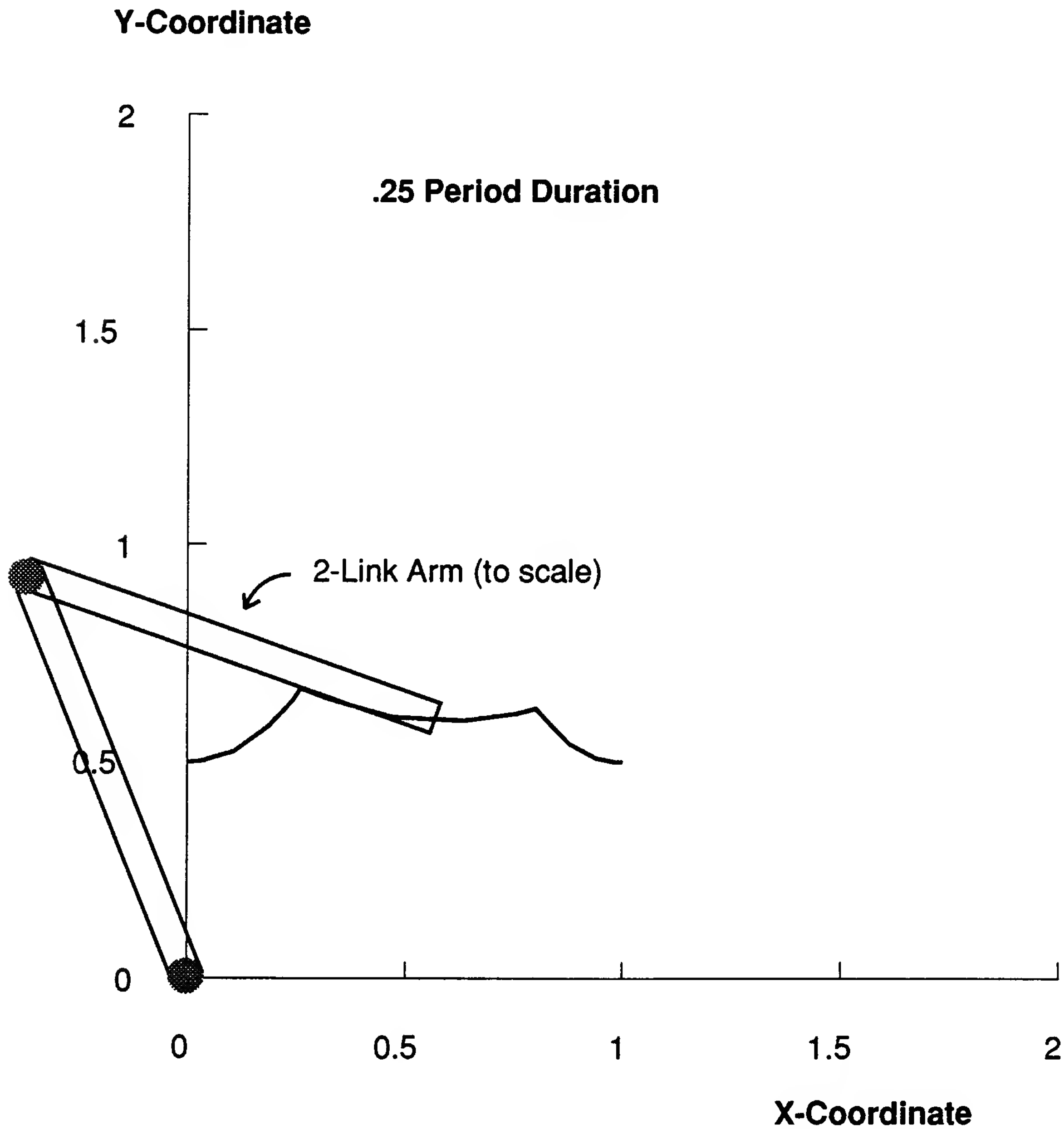


Figure 4.46: Cartesian trajectory comparison. The commanded trajectory is a fast versine in the X-direction with a constant .5 value commanded for the Y-direction. The The versine transitions from 0 to 1 in .25 periods. Note that this is not a time plot. The manipulator geometry is shown for a single point along the trajectory.

system to track the unshaped trajectory also degrades under these conditions. Figure 4.47 shows the response of a simple harmonic oscillator to the unshaped .1 period versine input (top) compared to the response of the same system to the shaped trajectory (bottom). Figure 4.48 shows the response of a simple harmonic oscillator to the unshaped .5 period versine input (top) compared to the response of the same system to the shaped trajectory (bottom).

The shaped trajectory becomes more jagged as the commanded trajectory becomes faster. However, the system tends to smooth the shaped trajectory in its response. The result is a response that has the same basic shape as the unshaped command without the vibration. It should be noted that the presence of the vibration alone in the response to an unshaped input degrades trajectory following considerably more than shaping does.

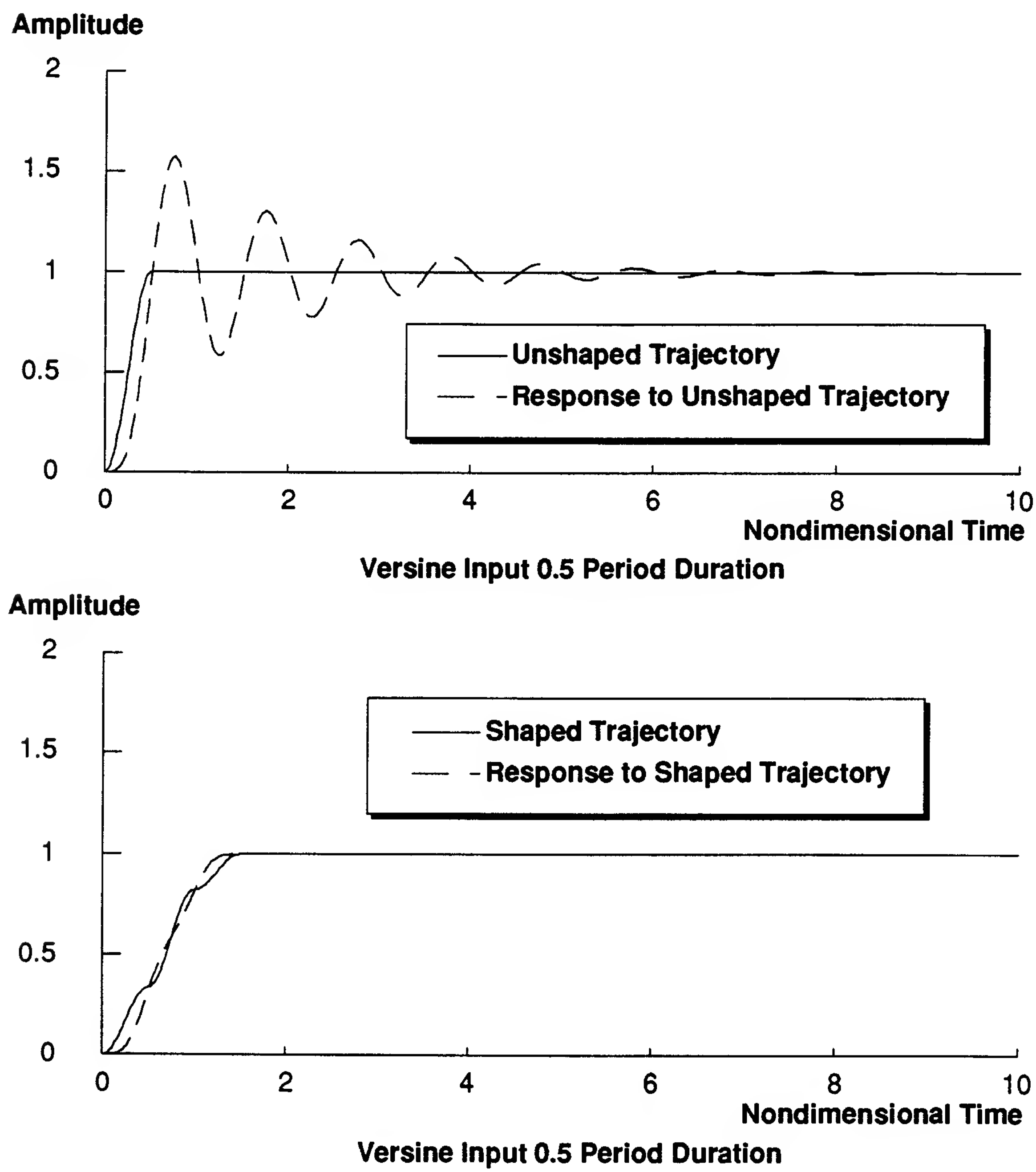


Figure 4.47: Trajectory response of a simple-harmonic oscillator. The input for both plots is a .5 period versine. The top plot shows the unshaped system response and the unshaped, commanded trajectory. The bottom plot shows the same input shaped with a three-impulse sequence superimposed over the system response. Note that the system tends to smooth the high frequency-content input of the bottom plot.

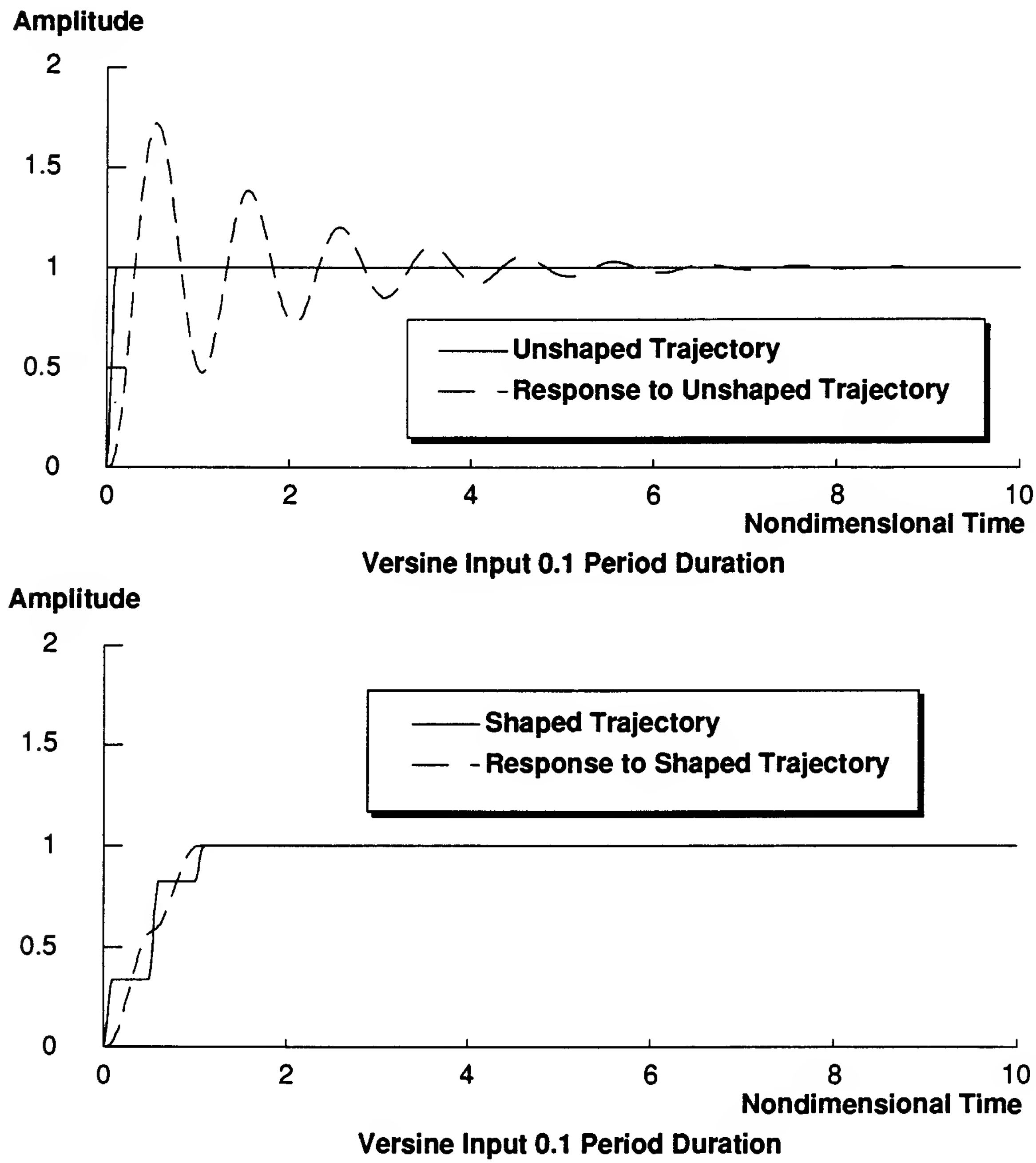


Figure 4.48: Trajectory response of a simple-harmonic oscillator. The input for both plots is a .1 period versine. The top plot shows the unshaped system response and the unshaped, commanded trajectory. The bottom plot shows the same input shaped with a three-impulse sequence superimposed over the system response. Note that the system tends to smooth the high frequency-content input of the bottom plot.

Hardware Experiments

Chapter 5

5.1 Design of a Flexible Test Fixture

A test machine was constructed for experimenting with various vibration-reducing strategies. This machine was used to verify the results of the shaping techniques presented, and to determine the overall feasibility of these techniques on real, non-idealized systems. This chapter describes the design of the test machine and then presents some results that were obtained.

5.1.1 Mechanical Hardware

The MIT Flexible Test Machine was designed and built by Andrew Christian and Neil Singer. It consists of a single rotational degree-of-freedom in the base; a rotary joint to which a slender link is attached; another rotary joint; and one additional slender link. At each of the three joints is a lumped rotational spring that can be adjusted for different stiffnesses. Each of the two thin links provides structural flexibility.

The primary goal of the MIT Flexible Test Machine design was to construct an instrumented laboratory machine that was representative of "real" machines. A secondary

requirement for the machine was to have large-amplitude, low-frequency, structural vibrational modes so that the oscillations are visual. A third requirement was to make the machine be similar in geometry to some class of industrial machines.

The first requirement meant that specialized hardware was not to be used — the components of the system were restricted to industrial components. The low natural frequency requirement creates design problems for machines that work in gravity environments (as on Earth). The endpoint deflection of the machine under the influence of gravity is directly related to the natural frequency of the machine by the following approximation:

$$\delta = \frac{g}{\omega^2} \tag{5.1}$$

where δ is the endpoint deflection due to gravity loading, g is the acceleration of gravity, and ω is the natural frequency of the system (see Blevins [18] pages 451-3). The details of this tradeoff and the MIT Flexible Test Machine design analyses can be found in [32].

The two rotary joints at the base use standard DC servomotors with a 10:1 timing belt reduction. The joint at the end of the first link uses a Samarium-Cobalt two piece torquer. The need for this one expensive motor was driven by the large-amplitude, low-frequency vibration requirement for the machine. An inexpensive motor at this location would have been heavier, and thus would have reduced the payload capacity of the arm and limited its visual impact by requiring that the vibrational deflections be reduced. Each joint has a brake and an optical shaft encoder.

5.1.2 Electronics

The computer and electronics configuration is a slightly customized version of the CONDOR system developed by Sundar Narasimhan and David Siegel [97,98]. This system consists of a Sun Microsystems 3/160 Computer connected to a VME backplane. Interface cards and 68020 microprocessors are inserted into the backplane and communicate

with the Sun using the CONDOR development system.

5.2 Hardware Results and Theory Verification

The shaping technique presented in Chapter 4 was first verified by using just the one base degree-of-freedom on the MIT Flexible Test Machine. A thin steel beam was mounted on the joint so that it could swing in the horizontal plane (figure 5.1). The beam was constructed from $\frac{1}{4}$ inch by $\frac{3}{4}$ inch by 24 inch long, ground 4140 half-hard steel. A large mass was cantelevered at 19 inches from the rigid beam supports. Adjustable weights could also be placed at the end of the beam so that the natural frequency of the system could be easily varied.

A simple PD Control loop was implemented by graduate student, Erik Vaaler. A linear-quadratic gaussian observer was designed. The noise that was assumed was due to discretization from the encoder. Figure 5.2 shows a step response of the table when no flexible members are attached to it. This figure demonstrates that the flexibility introduced into the system is from the structural flexibility and not the controller. The controller has well damped poles at approximately 15 hz. — well above the 2 - 7 hz modes of the flexible beams.

5.2.1 Robustness Verification

The first experiment was a verification of the robustness of the new technique. Normally the system natural frequencies would be estimated and the sequence would be designed for those frequencies. However, because of uncertainties in the parameters of real systems, these frequencies may no longer be the ones that were estimated. To simulate this uncertainty and verify the robustness of the new shaping technique, the shaping sequences were intensionally designed for frequencies that were different from the measured frequency of the beam system. In a "real" system, the system would be changing

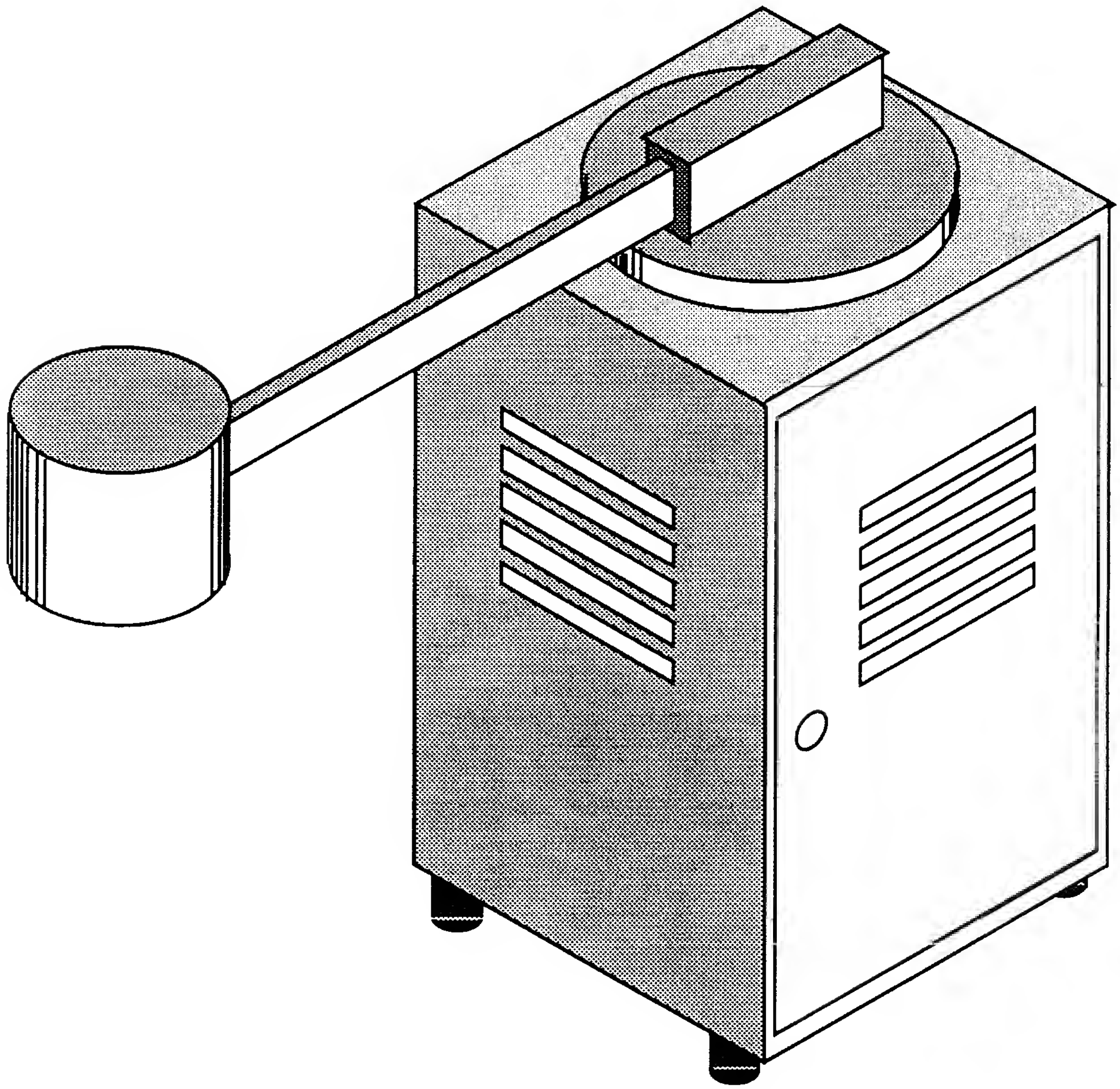


Figure 5.1: Sketch of the beam setup that was used for the hardware experiments.

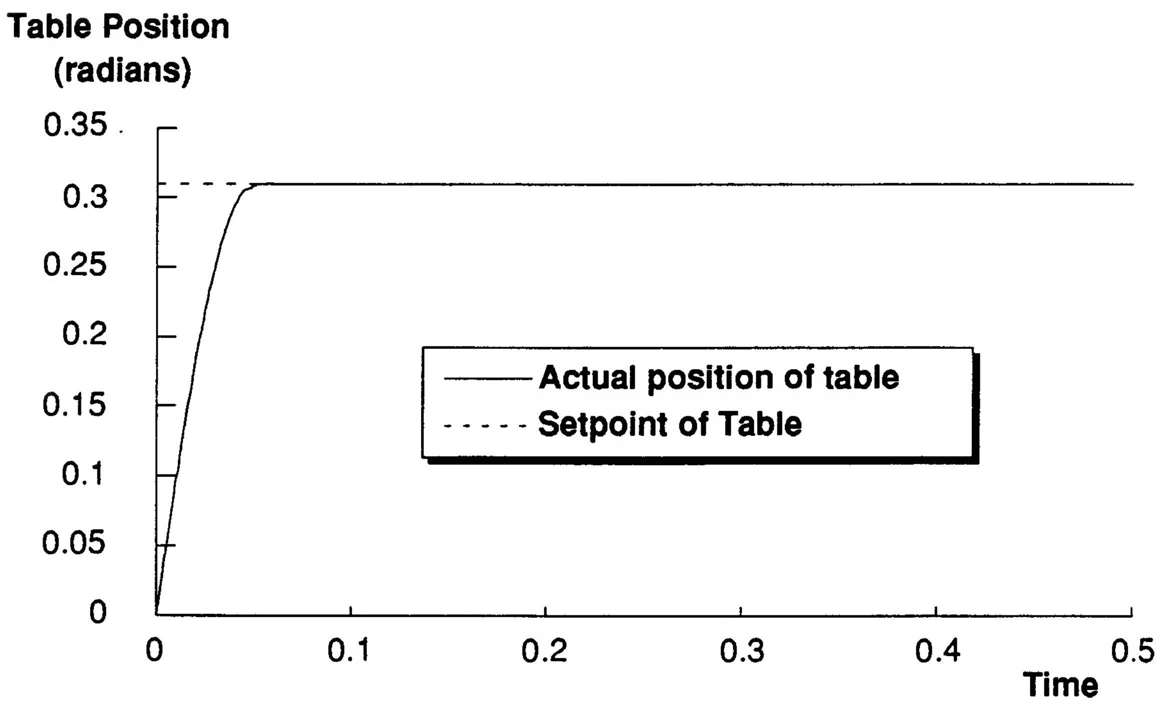


Figure 5.2: Step response of the base joint of the MIT Flexible Test Machine. The motion is rotary in the horizontal plane. The flexible beam is not mounted on the table. This plot shows that the flexibility present in the later moves is from the structure and not the controller.

and the shaping sequence would be fixed. However, it was easier to experimentally collect data when the system was left unaltered and the sequences were varied over a large range in software. The two situations are entirely equivalent because the type of robustness that is of interest is the ability of the sequences to reduce vibration when the system has different parameters from those that were expected when the sequence was designed. For visual demonstrations (as opposed to data collection), the mass at the end of the beam is varied while the sequences are kept fixed.

Two-, three-, and four-impulse filters were generated for a fixed, 2.45 hz system. Step inputs were sent to the system while the amplitude of the vibration was measured at the encoder. The vibration was due to the combined flex in the steel beam and stiffness in the joint servos. The motion of the end mass feeds back into the joint and causes an oscillation which can be monitored at the encoder. The higher modes of the beam were ignored because none of them appeared significant in the data.

Figures 5.3 5.4 and 5.5 compare the hardware results obtained from the MIT Flexible Test Machine to the theoretical plots presented in Chapter 4. The theoretical curves on the plots of figures 5.3 5.4 and 5.5 are plots of:

$$\text{Vibration Error} = \frac{\sqrt{V_1^2 + V_2^2}}{8.5} \tag{5.2}$$

with the V_1 and V_2 given in equation 3.6, and with the impulse sequence normalized to unity (the sum of the A_j is one). The factor of 8.5 is the ratio of the joint stiffness to the beam stiffness. This factor is included because the vibration is being measured at the encoder. Any vibration at the endpoint will appear at the encoder scaled down by a factor of 8.5. Measuring the motion of the mass at the end of the beam is more difficult than measuring the encoder data and requires special hardware. The decision was made to tradeoff the signal strength for ease of measurement and scale the theoretical curves accordingly.

Figure 5.3 shows the lack of robustness of the two impulse sequence. Figure 5.4

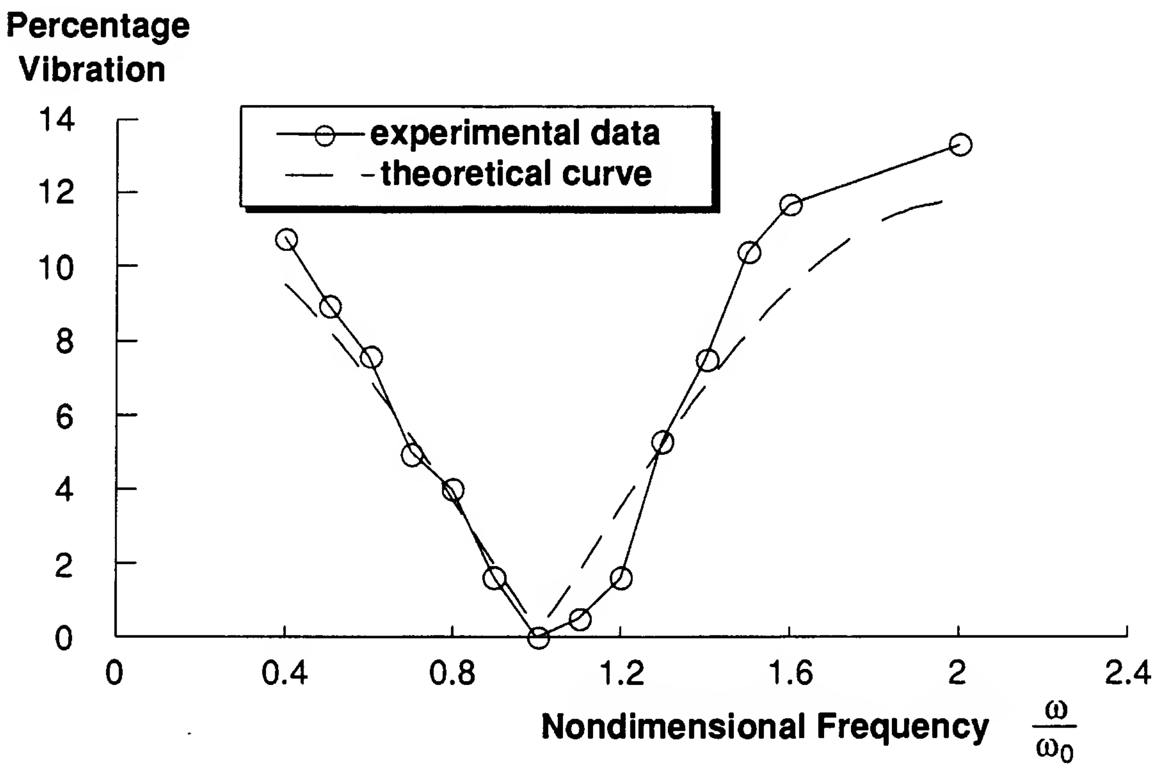


Figure 5.3: Comparison of hardware vibration error vs system natural frequency to theory (two-impulse sequence). The test system was fixed with ω_0 = 2.45 hz. The two-impulse shaping sequence was varied to intensionally create an error (and residual vibration).

demonstrates the effect of using a three impulse sequence and figure 5.5 shows the effect of using a four impulse sequence.

Figure 5.6 shows a time response of the single mass-beam system using a two impulse sequence. The top plot shows the response when the system and sequence design frequency are the same (as close as possible using real hardware). The bottom plot shows the response when the sequence is designed for a frequency that is 40% higher than the measured frequency of the beam-system. For reference, the unaltered step response of the system (relying on damping alone) is shown superimposed on these plots. Figure 5.7 presents the same set of plots using a three-impulse shaping sequence. Figure 5.8 presents the same set of plots using a four-impulse shaping sequence.

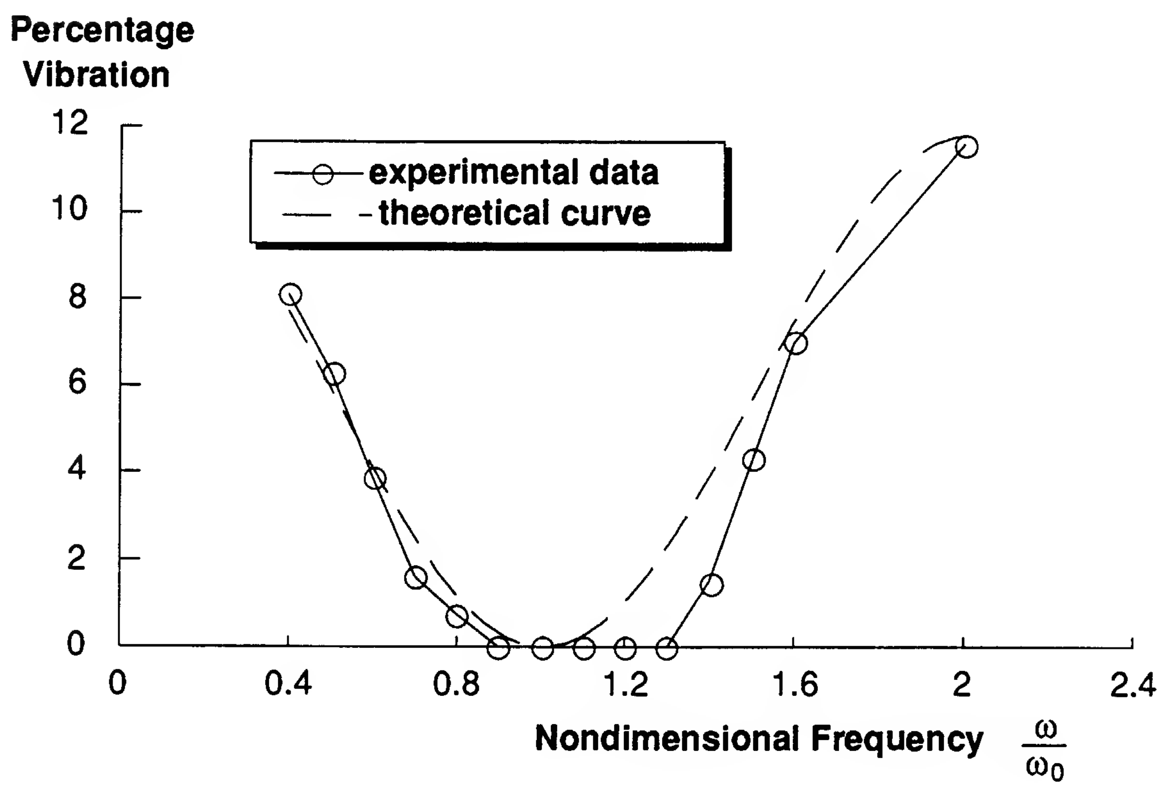


Figure 5.4: Comparison of hardware vibration error vs system natural frequency to theory (three-impulse sequence). The test system was fixed with $\omega_0 = 2.45$ hz. The three-impulse shaping sequence was varied to intensionally create an error (and residual vibration).

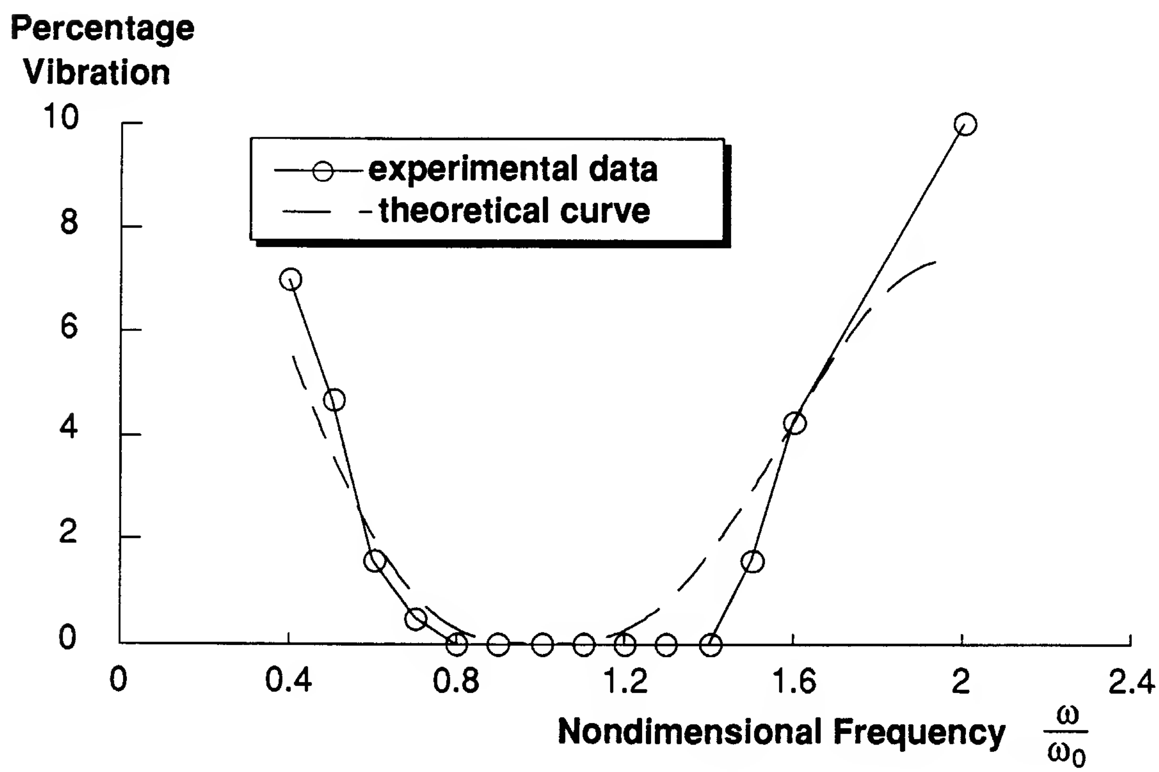


Figure 5.5: Comparison of hardware vibration error vs system natural frequency to theory (four-impulse sequence). The test system was fixed with $\omega_0 = 2.45$ hz. The four-impulse shaping sequence was varied to intensionally create an error (and residual vibration).

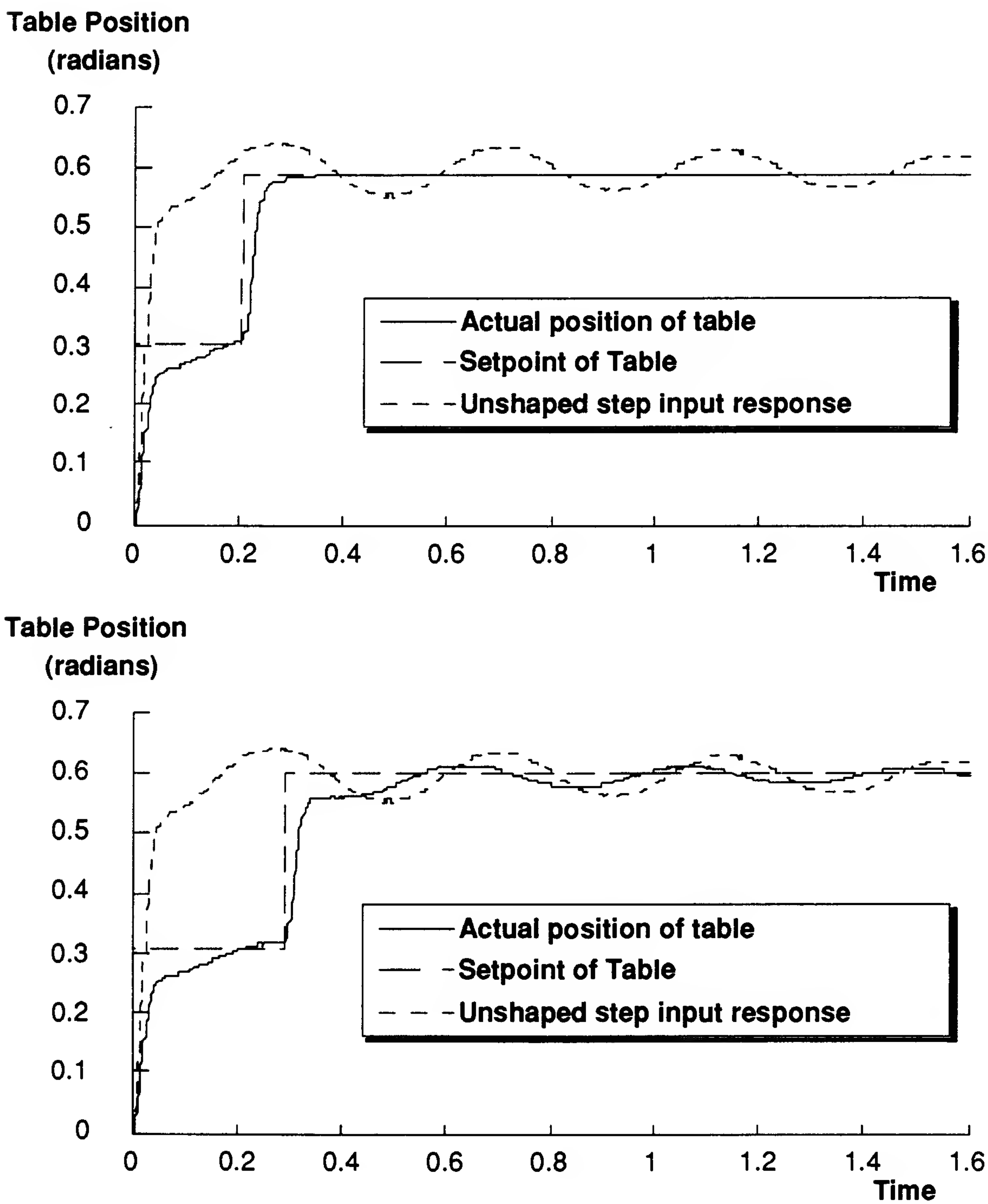
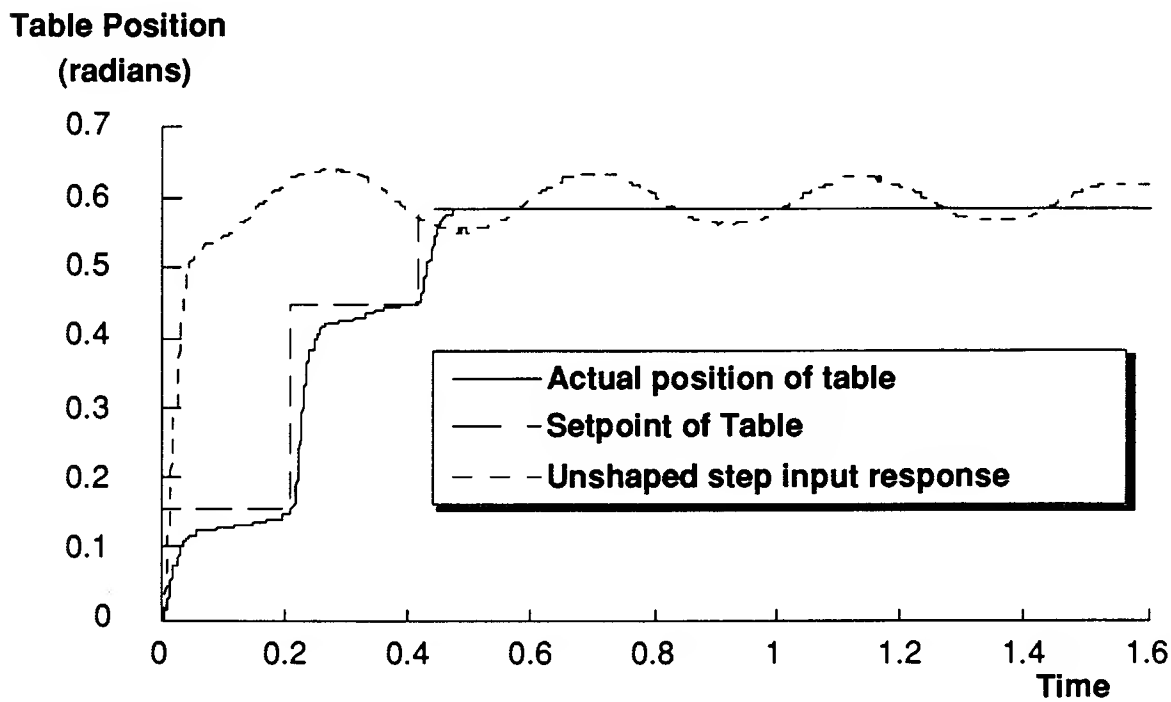


Figure 5.6: Step response of the hardware experimental system shaped with a two impulse sequence. The top plot shows the response when the sequence design frequency is the system natural frequency. The bottom plot shows the response when the filter design frequency is 40% higher than the system natural frequency. The setpoint to the shaping routine is a step to .6 radians. The dashed line is the signal (after shaping) that is sent to the servo.



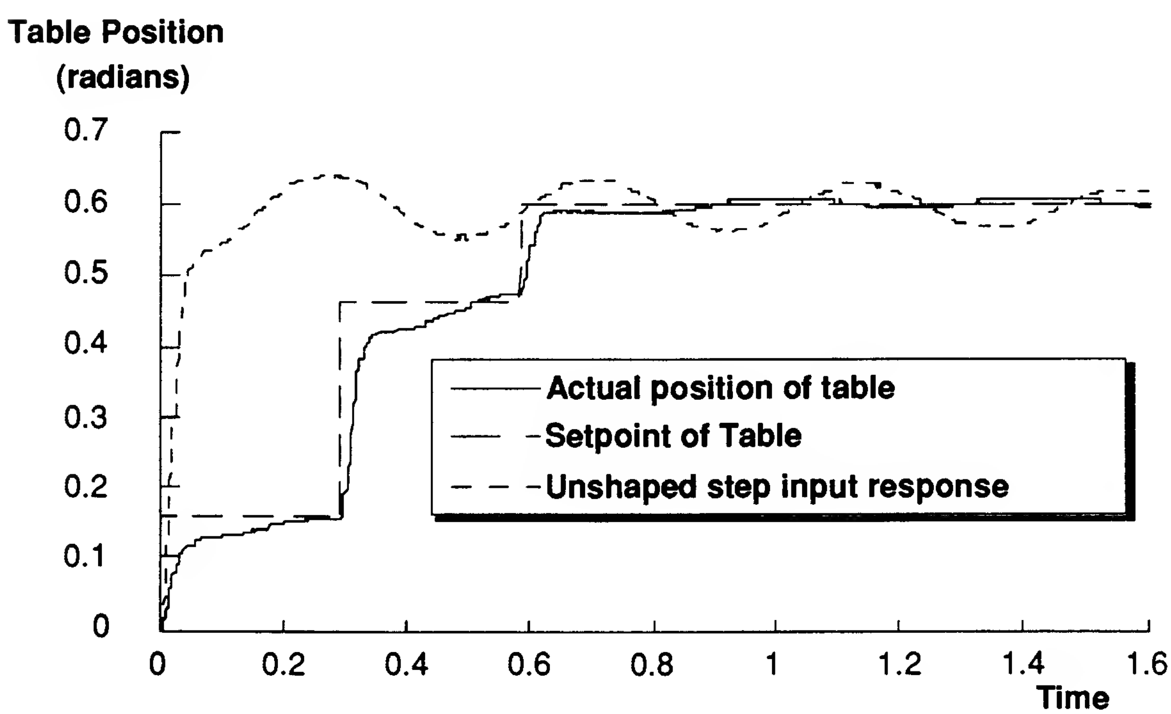


Figure 5.7: Step response of the hardware experimental system shaped with a three impulse sequence. The top plot shows the response when the sequence design frequency is the system natural frequency. The bottom plot shows the response when the filter design frequency is 40% higher than the system natural frequency. The setpoint to the shaping routine is a step to .6 radians. The dashed line is the signal (after shaping) that is sent to the servo.

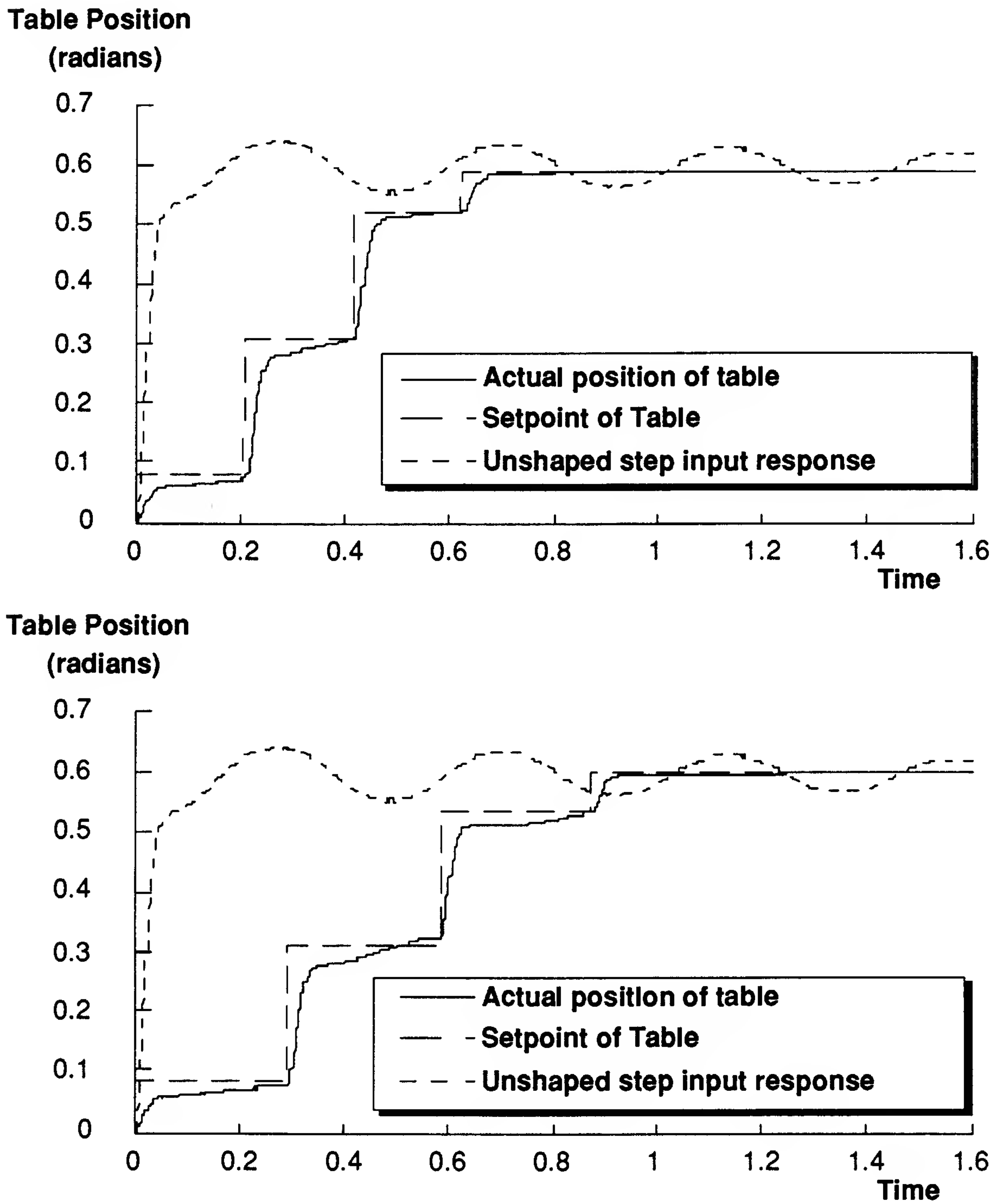


Figure 5.8: Step response of the hardware experimental system shaped with a four impulse sequence. The top plot shows the response when the sequence design frequency is the system natural frequency. The bottom plot shows the response when the filter design frequency is 40% higher than the system natural frequency. The setpoint to the shaping routine is a step to .6 radians. The dashed line is the signal (after shaping) that is sent to the servo.

5.2.2 Effectiveness on Arbitrary Inputs

The next experiment involved generating arbitrary command inputs from a human operator and shaping them in real time. A potentiometer was used as a joystick so the operator of the machine could send position setpoints to the closed-loop servo on the one degree-of-freedom system. Figures 5.9 and 5.10 show a comparison between one run generated by sending the unshaped joystick commands to the servo, and another run generated by first shaping the joystick signal. For each run, the joystick command is shown as a dotted line. The response shown is that of the rotary table at the motor encoder. The vibrations shown in the upper plot of figure 5.9 correspond to a 7 inch amplitude oscillation at the mass on the end of the 19 inch long beam. The joystick inputs to the system had to be moderated when shaping was not used so that the beam would not plastically deform. The data shown on the plots are the reduced-amplitude encoder positions (reduced by a factor of 8.5).

5.2.3 Multiple Mode Operation Verification

In order to verify the extensions of the new input shaping technique to multiple mode systems, the hardware was slightly modified. A second beam was attached to the hub so that two structural modes and one servo mode were present. Figure 5.11 shows the configuration used for the experiments in this section. Figure 5.12 shows the response of the two-beam system to an unaltered step input. Both modes appear in the output response of the system. Figure 5.13 shows the effect of shaping to remove only one of the two modes of the two-beam experiment. The top plot used a sequence designed for the 2.45 hz mode. The bottom plot used a sequence tuned for the 6.4 hz second-mode. Figure 5.14 shows the effect of shaping for both modes. The shaping sequence for both modes was generated by convolving the three-impulse seuqence designed for the 2.45 hz. mode with the three-impulse sequence designed for the 6.4 hz. mode. The result-

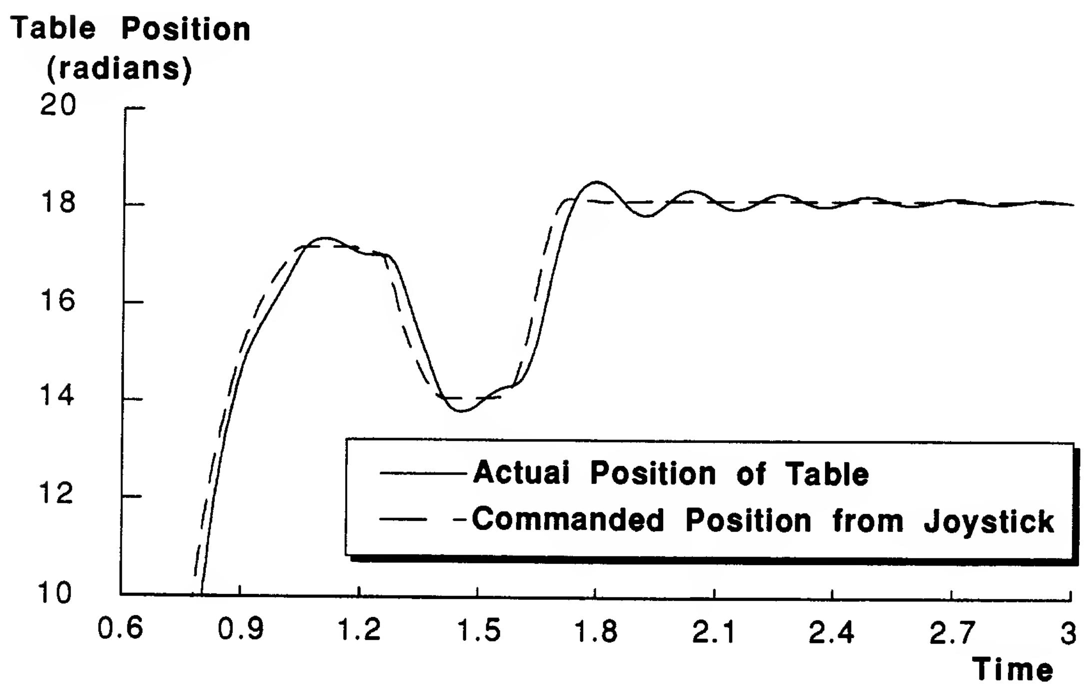


Figure 5.9: Response of the hardware system to arbitrary inputs. The solid trace is the response of the experimental system to a joystick input. In this run the command is not shaped.

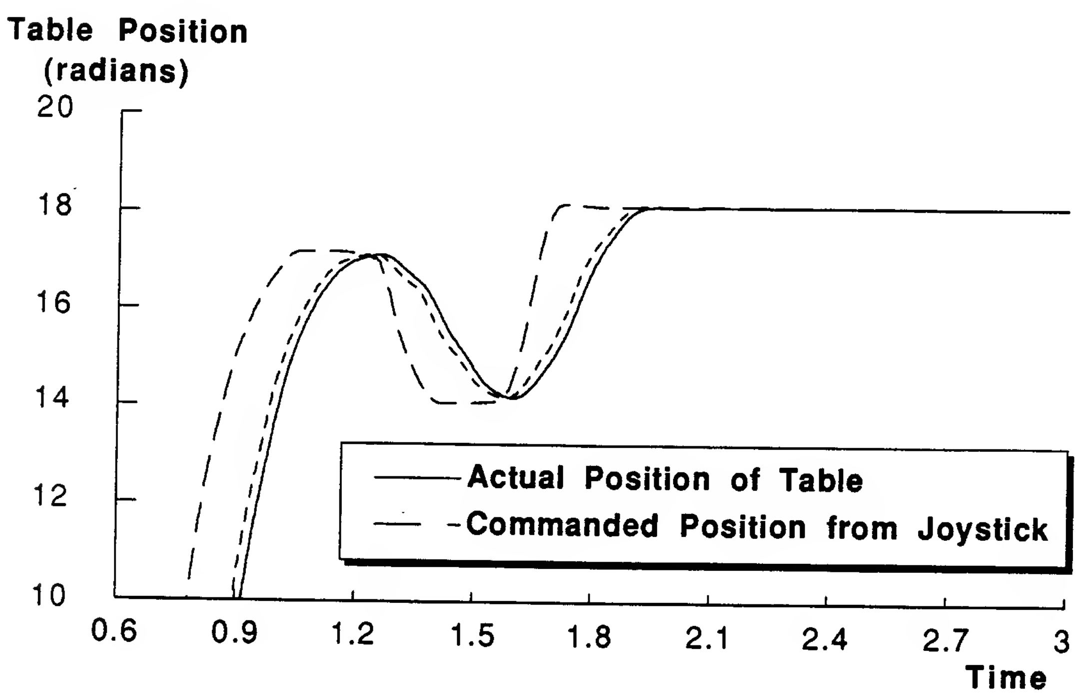


Figure 5.10: Response of the hardware system to arbitrary inputs. The solid trace is the response of the experimental system to a joystick input shaped with a three-impulse sequence. The shaped command is shown with a short-dashed line. Note the time delay is one period of the system.

ing nine-impulse sequence was slightly longer than necessary. The shortest sequences are generated using a nonlinear equation solving routine (Chapter 3 and section 4.6), however, for simplicity, the convolution approach was used in this experiment.

5.2.4 Negative Impulses — Shorter Sequences

Figure 5.16 shows the effect of using a shorter sequence for shaping the system input to the single beam system. A negative sequence (shown in figure 5.15) .27 seconds in length was used to shape inputs to the single-mode system. The use of this sequence saved 30% in the shaping delay time. The sequence was generated by convolving two of the negative sequences shown in figure 4.28 together in order to form a robust sequence. (Alternatively, the equations presented in chapter 4 could have been solved for a similar sequence.) Figure 5.17 shows hardware vibration measurements as the sequence design frequency is varied away from the natural frequency of the system. This curve is shown on top of the theoretical vibration error curve for a three-impulse sequence. Note that the negative sequence was designed so that it would meet the same robustness constraints as the three-impulse sequence.

5.2.5 Sequences that Meet Multiple Constraints

Section 3.4 presented a formulation for including additional constraints in generating sequences so that more robustness could be obtained. This section discusses the implementation of such a sequence on the MIT Flexible Test Machine. The sequence that was developed was intended to be robust over a braod range (between one and five hertz). The vibration error and derivative were set to zero at 1.5 hertz and 3.5 hertz. The vibration error, only, was set to zero at 2.45 hertz. The sequence that is shown in figure 5.19 was generated by convolving three individual sequences that each satisfy one of the above-mentioned constraints. This method was more costly in terms of time

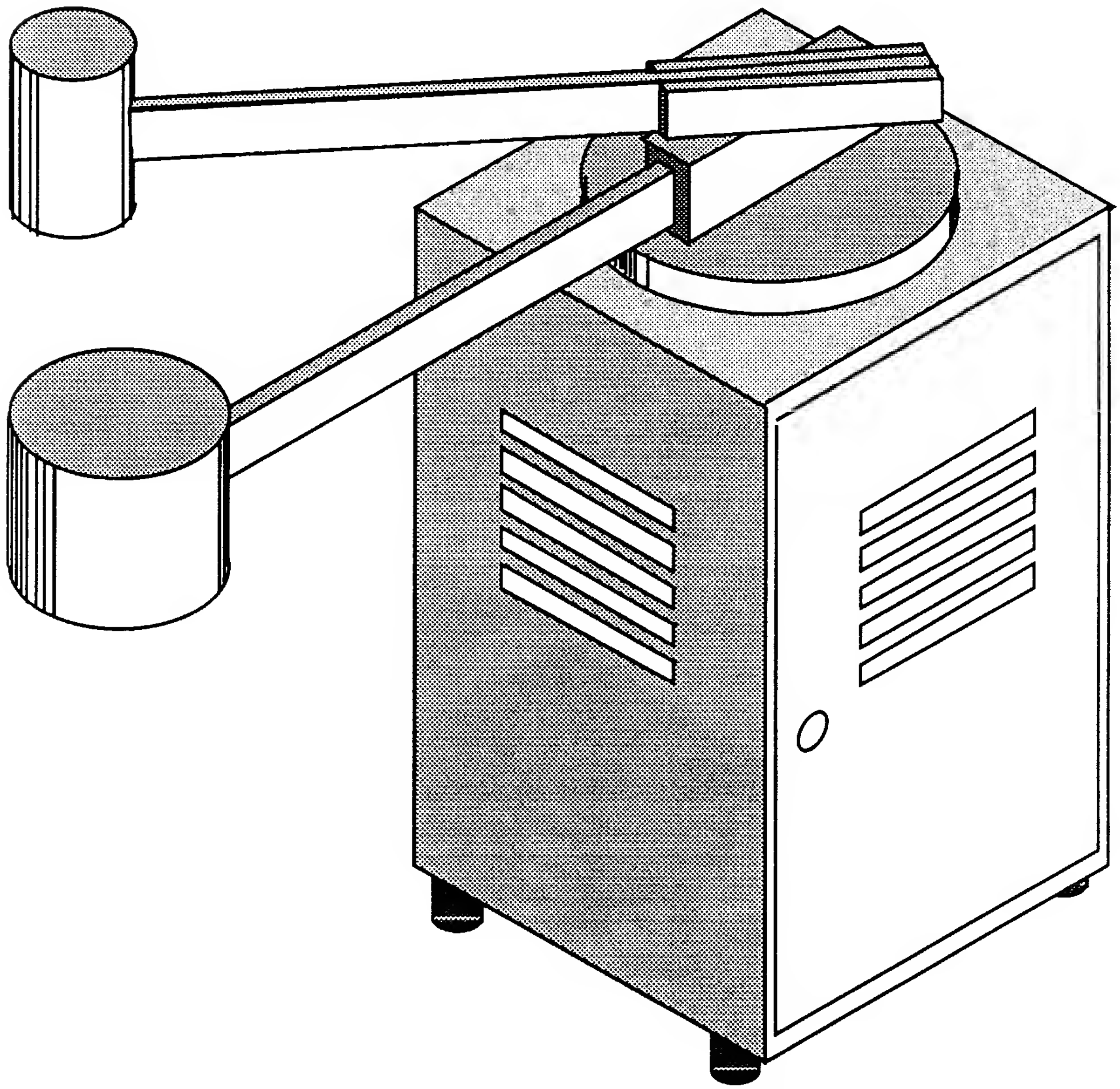


Figure 5.11: Sketch of the two-beam setup that was used for the multiple-mode hardware experiments.

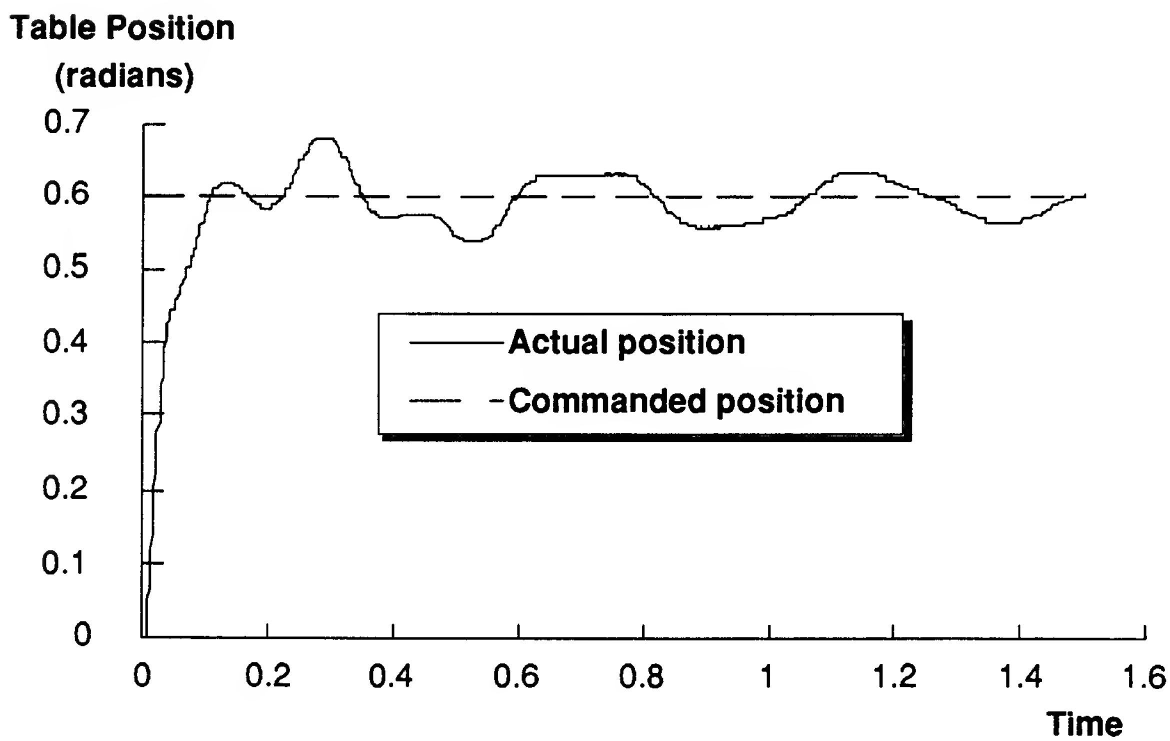


Figure 5.12: The two beam system commanded with an unshaped step input. This plot is for comparison with the next three figures.

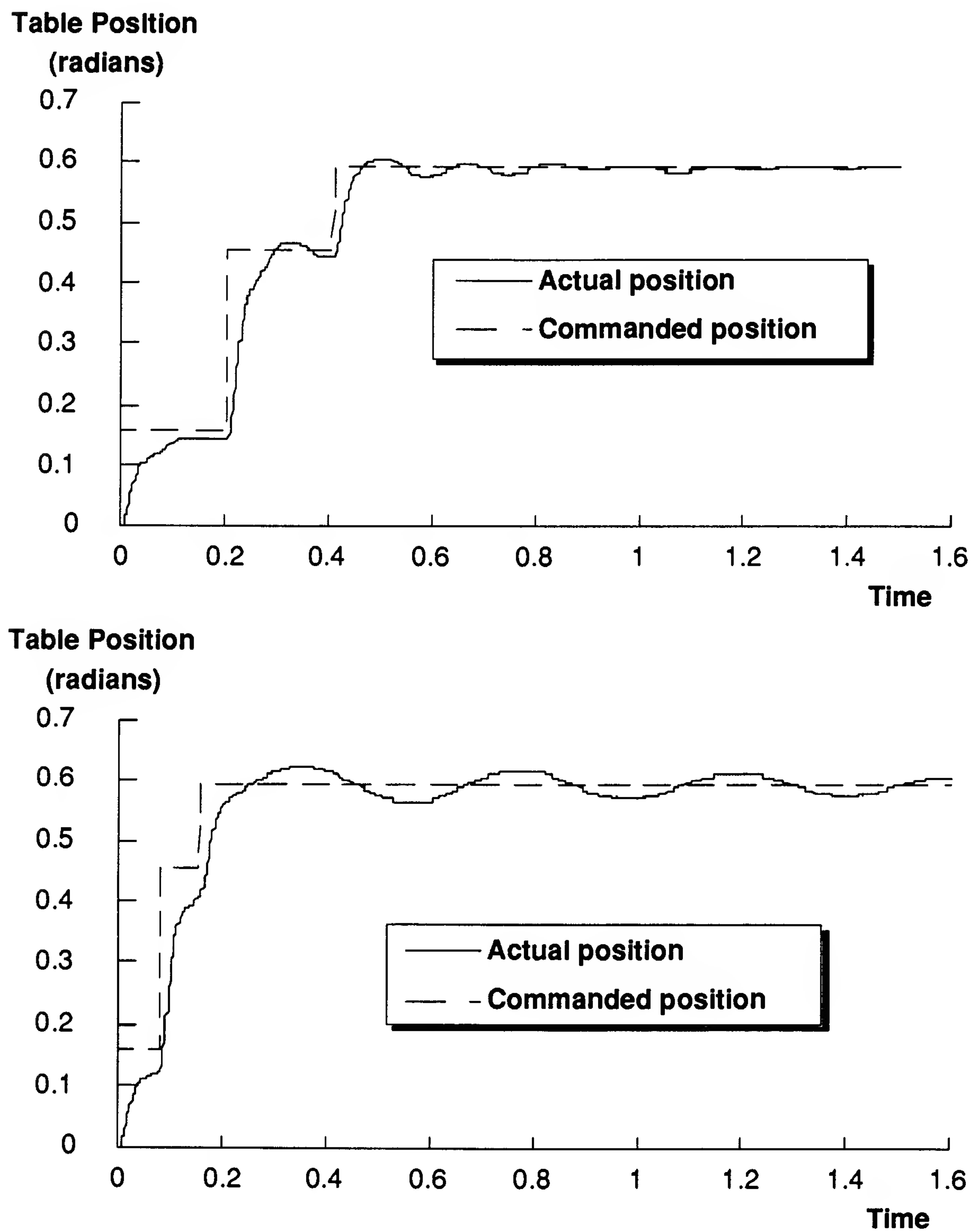


Figure 5.13: The two beam system shaped to remove just one mode. The top plot shows the response with just the low frequency mode removed. The bottom plot shows the response with just the high frequency mode removed. The setpoint to the shaping routine is a step to .6 radians. The dashed line is the signal (after shaping) that is sent to the servo.

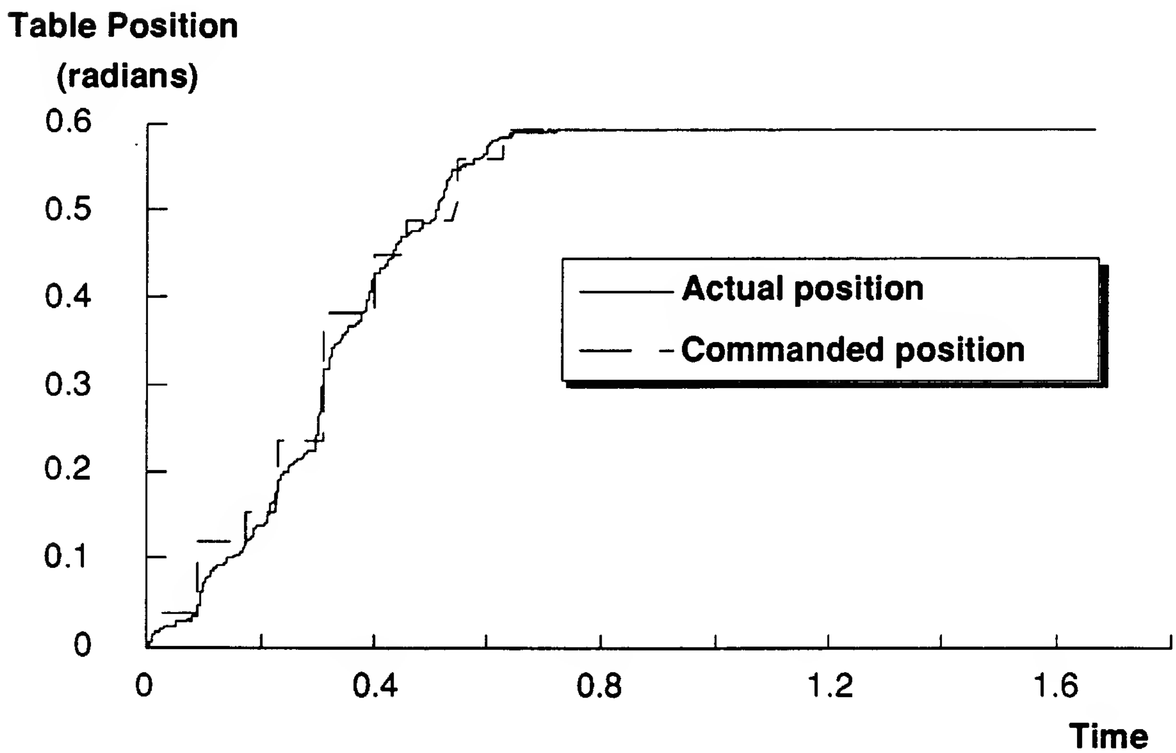


Figure 5.14: The two beam system shaped to remove both modes. The setpoint to the shaping routine is a step to .6 radians. The dashed line is the signal (after shaping) that is sent to the servo.

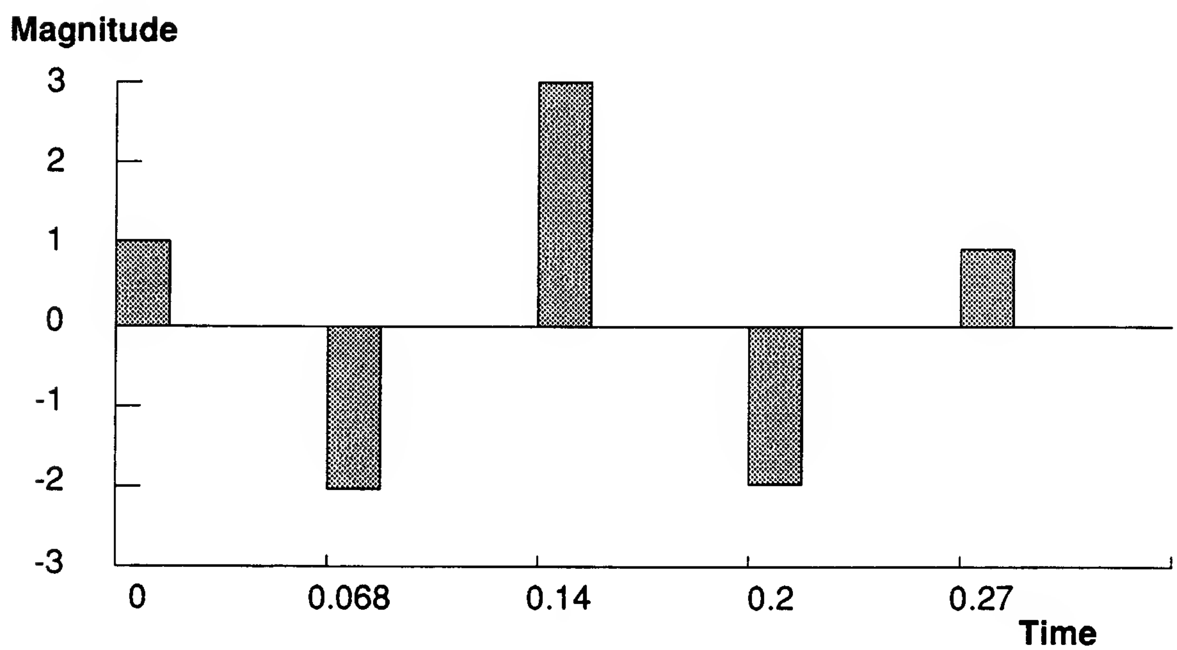


Figure 5.15: A shorter, negative sequence is designed for the single-beam system. This sequence is compared to the three-impulse sequence. Both sequences meet the same mathematical constraints. The negative sequence saves time at the expense of actuator effort.

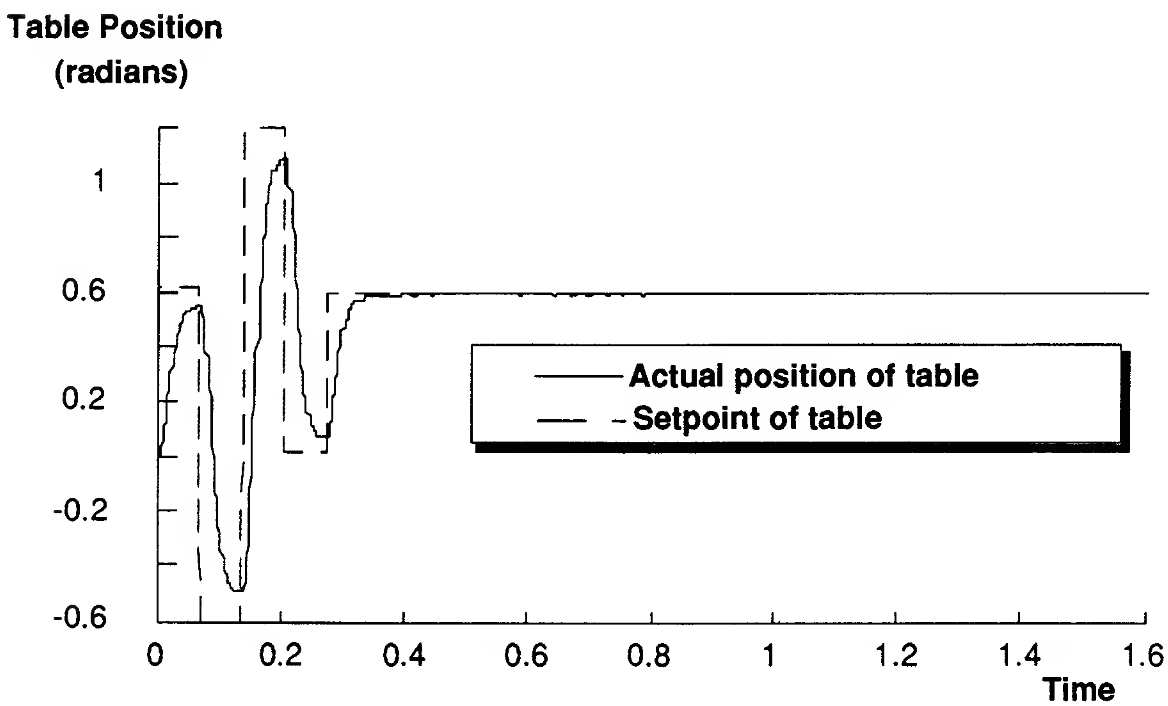


Figure 5.16: The single beam system is commanded with a shorter sequence. The sequence is shortened by using both positive and negative impulses. The setpoint to the shaping routine is a step to .6 radians. The dashed line is the signal (after shaping) that is sent to the servo.

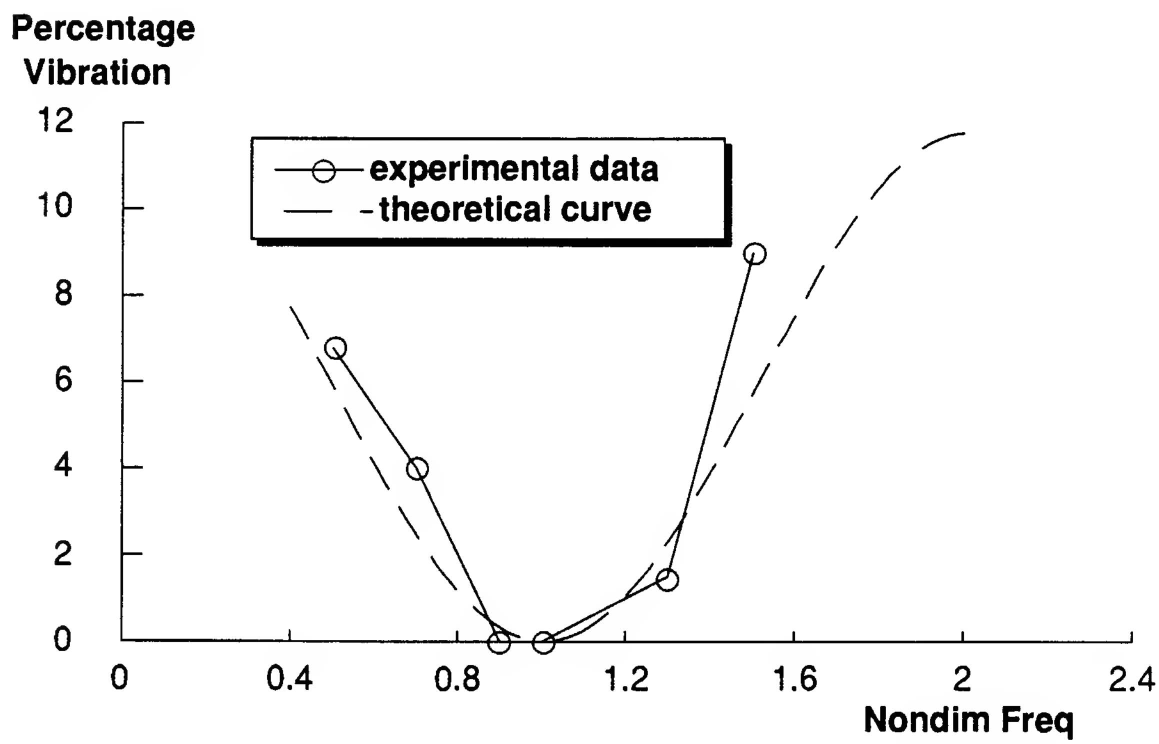


Figure 5.17: Comparison of hardware vibration error vs system natural frequency to theory (negative impulse sequence). The negative impulse sequence shown in figure 5.15 was used to shape inputs to the hardware system. The robustness curve is compared to that of the three-impulse sequence to show that the same constraints are satisfied.

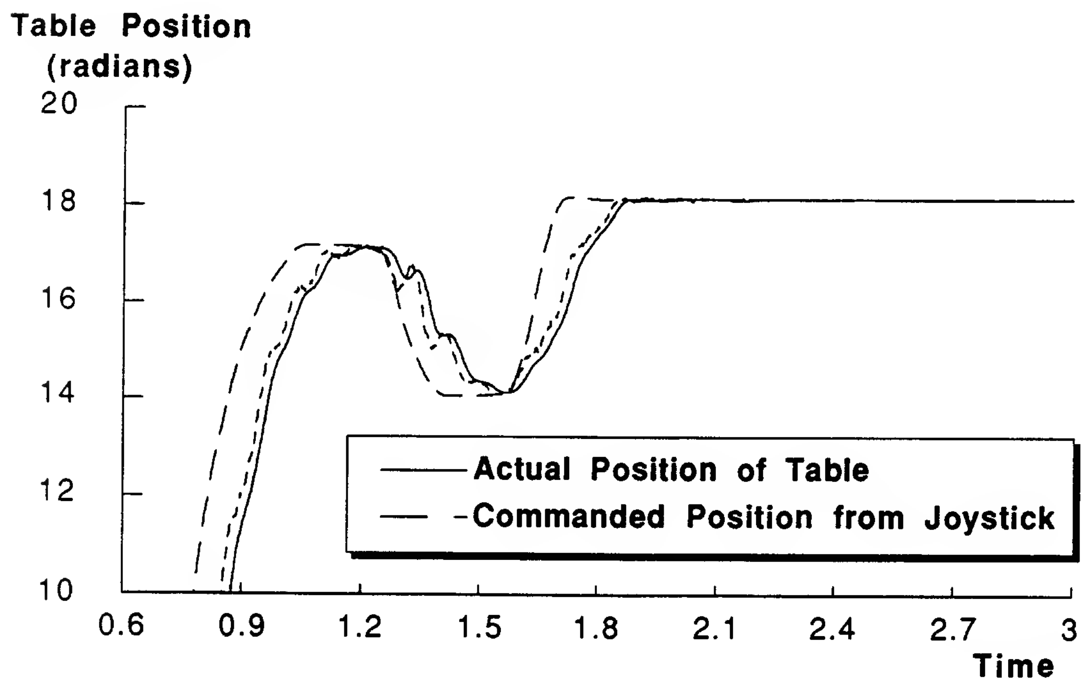
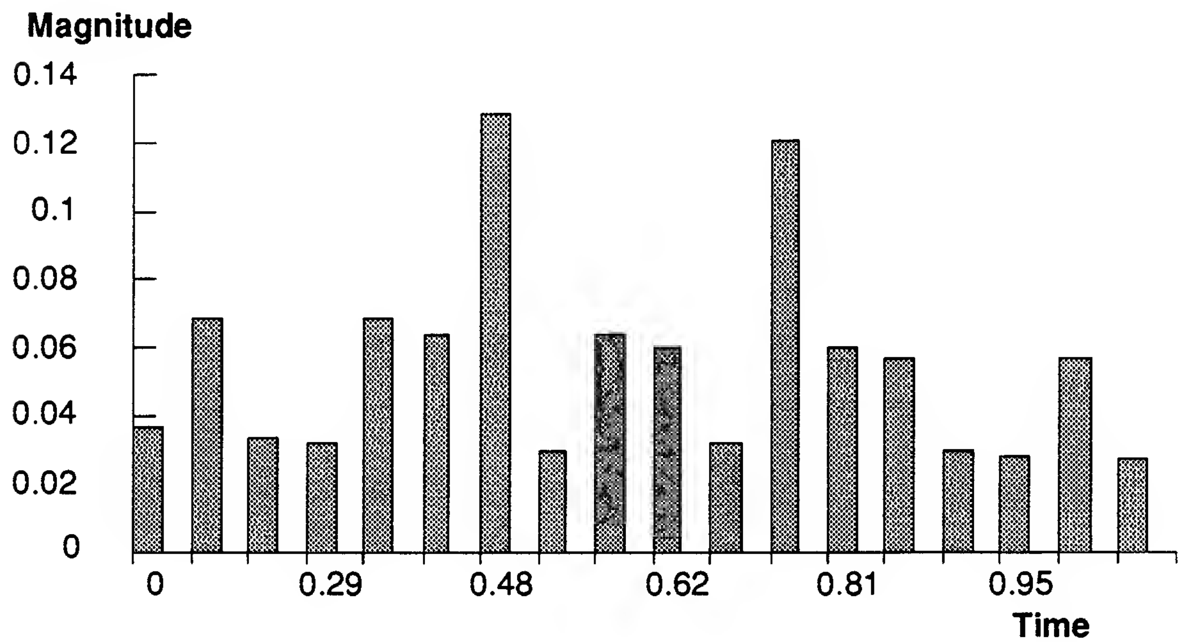


Figure 5.18: Negative sequence effect on joystick inputs. The solid trace is the response of the experimental system to a joystick input shaped with the negative impulse sequence shown in 5.15. The shaped command is shown with a short-dashed line. Note the time delay is two-thirds of one period of the system. The joystick input is the same as that used in figure 5.9

duration of the input, however, generation of the sequence is simplified.

Figure 5.20 shows a time response of the mass-beam experimental system making a step move that was shaped with the sequence shown in figure 5.19.



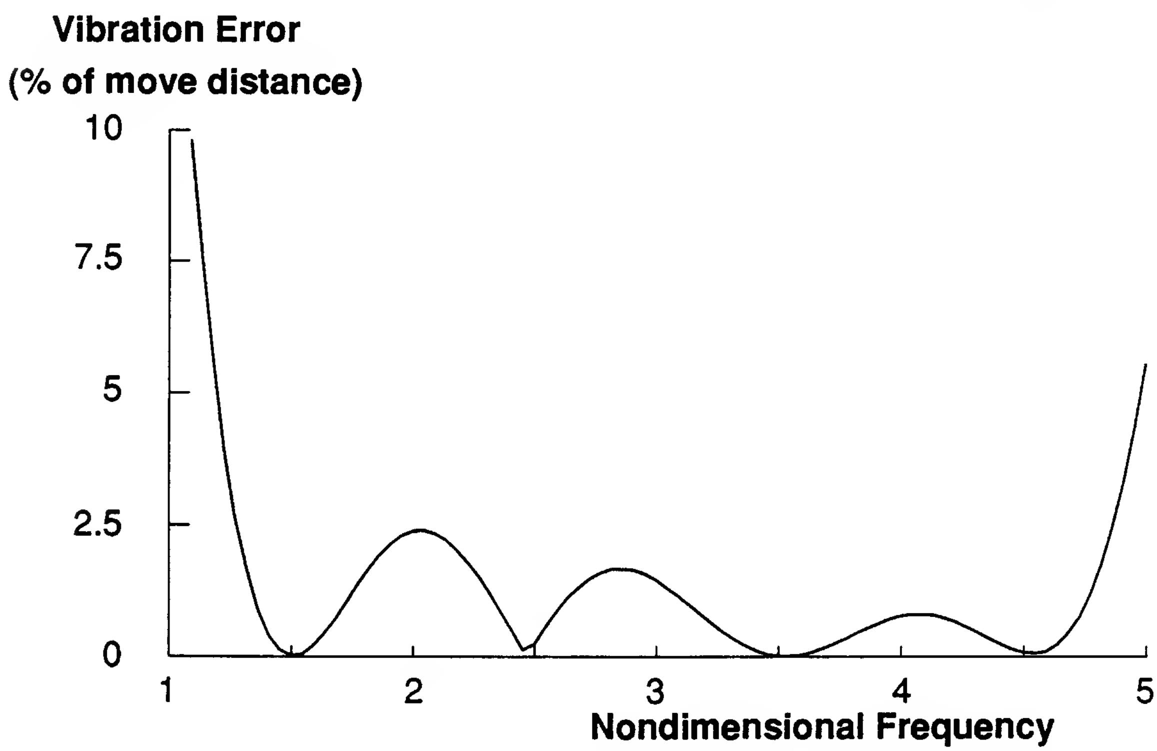


Figure 5.19: Top: Eighteen impulse sequence used for shaping inputs to the MIT Flexible Test Machine. This sequence was generated by convolving a three-impulse sequence designed for 3.5 hz; a two-impulse sequence designed for 2.45 hz; and a three-impulse sequence designed for 1.5 hz. The resulting sequence is long, however, it is robust for large frequency changes in the system. Bottom: A plot of vibration error as a function of system natural frequency.

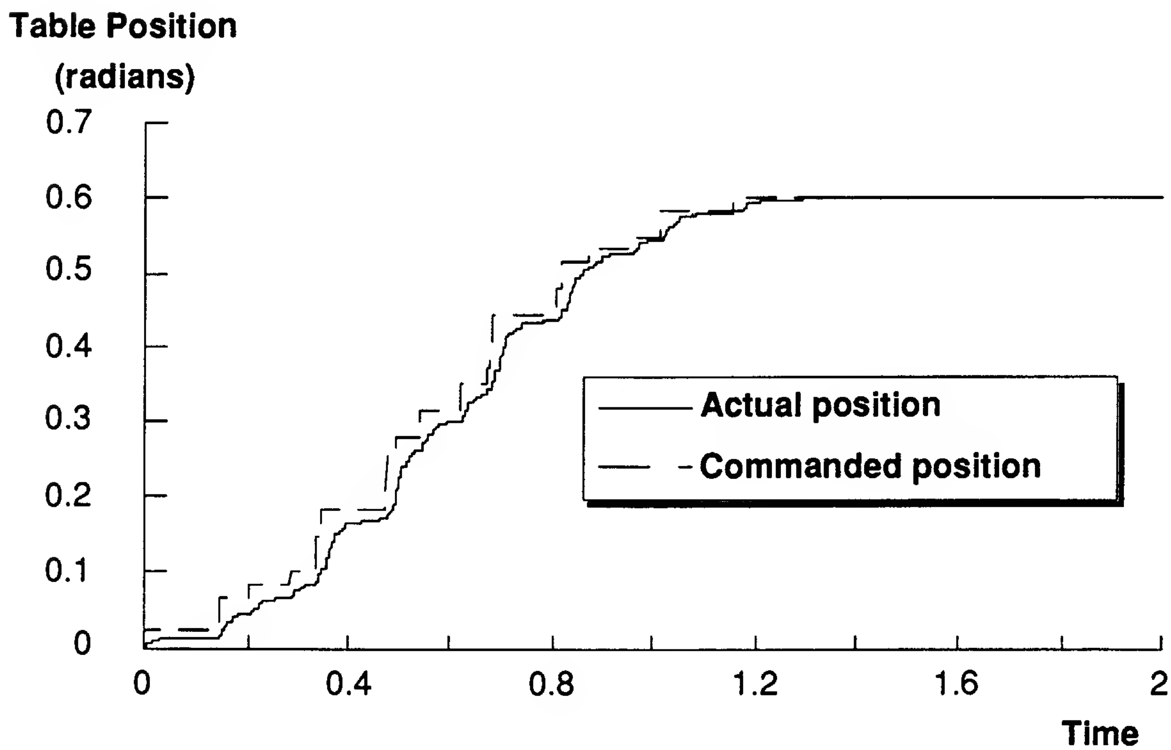


Figure 5.20: Response of the experimental system using the sequence shown in figure 5.19.

Evaluation of the Time Domain Shaping Method

Chapter 6

6.1 Introduction

The technique presented in the previous chapter essentially results in a finite impulse response (FIR) filter. These filters, however, are unlike filters produced by conventional design methods because they are not designed exclusively in the frequency domain. The constraint equations that were used to derive the filters are essentially time domain constraints. The impulse sequences can be considered "notch" filters in a decaying sinusoidal domain — Just as a frequency notch filter removes a range of frequency components from a signal, the new sequences remove a range of decaying sinusoidal responses. This effect makes them particularly useful for preshaping signals to vibratory mechanical systems.

Three criteria will be used to evaluate the various techniques. First, the impulse response duration is important. The time required for the system to complete a move depends on the duration of the impulse response. The impulse response is a lower bound on the move time. If the shaping technique were perfect, the system would make any

requested move with a fixed time penalty of the length of the filter's impulse response. The second criterion is a measure of the residual vibration after the filter (or shaping method) is finished. The third criterion is the robustness of the shaping technique to uncertainties in the system.

6.1.1 Tests Performed

In this chapter, each of the filters described will be used to shape the inputs to a simple harmonic oscillator with one natural frequency (set to be one hertz). In order to evaluate the various filters, two plots will be used. Two step responses will be shown for each filter. One is the response of the test system when it has the exact natural frequency that was assumed during the filter design process. The second is the step response of the test system when its natural frequency is set 15% lower than expected. 15% was chosen so that the robustness comparison is made when a significant, but realistic error is induced. The frequency is lower than expected in order to fairly evaluate lowpass filters. Lowpass filters, by definition work better as the frequency of the system is increased. However, a significant time penalty is incurred by making the cutoff frequency of the lowpass filter too low. Therefore, there is an incentive to make the filter cutoff as high as possible, thus overestimates of the natural frequency will be most common. The results of this section will show that lowpass filters perform significantly worse than notch filters, therefore, this tradeoff will not be an issue. Both systems have no damping so that two quantities can be observed — the time at which the filter finishes, and the residual vibration amplitude. Damping is removed in order to isolate the performance of the filters.

This chapter will demonstrate that filters designed in the frequency domain will result in large move time penalties and/or significant residual vibration compared to moves performed using the shaping methods of chapter 3 and 4. Systematically, a range

of filter types will be addressed.

In the following sections, results obtained using a series of filtering techniques are presented as step response plots. The added lines on the plots indicate the performance specifications that are tabulated in the summary. On the left-hand axis the residual vibration for a system that has the expected natural frequency (exact) is expressed as a percentage of the move distance. The two lines indicate the absolute magnitude of the residual vibration for this response (the solid curve). On the right-hand axis the residual vibration for a system that has a natural frequency that is 15% lower than expected is expressed as a percentage of the move distance. The two lines on this side indicate the absolute magnitude of the residual vibration for this response (the dotted curve). For both responses, the filter "ends" at the time indicated by the open circle on the axis. The "end" of the filter was determined as the time at which the filter output stays within 5% of its steady-state final value. The residual vibration is the magnitude of the oscillation after this time. For some of the filters, the end time is longer than the 10 second window shown and is, therefore, off the graph. This situation is indicated with an arrow next to the circle.

6.1.2 Three-Impulse Sequence

Figure 6.1 shows a step response of a system whose input is shaped with a three-impulse sequence. Two responses are shown. The first, in black, shows the response of a system that has the natural frequency that was expected in designing the shaping sequence. The second response, in gray, is that of a system with a natural frequency 15% lower than the design frequency. Note that for a significant error in the system parameters, very little vibration penalty is incurred.

6.1.3 Four-Impulse Sequence

Figure 6.1 shows a step response of a system whose input is shaped with a four-impulse sequence. Two responses are shown. The first, in black, shows the response of a system that has the natural frequency that was expected in designing the shaping sequence. The second response, in gray, is that of a system with a natural frequency 15% lower than the design frequency. Note that the four-impulse sequence results in less residual vibration for the same amount of error in the system natural frequency (as compared to the three-impulse sequence). The response time, however, is longer for the four-impulse sequence.

6.2 Lowpass Filters

6.2.1 Ideal Lowpass Filters

Ideal lowpass filters have the rectangular magnitude of freqeuncy response shown in figure 6.2. This ideal filter is physically unrealizable because the impulse response continues indefinitely. Truncation of the impulse response limits the desired frequency response by what is referred to as Gibbs Phenomenon [105] (pp. 239–41). Figure 6.2 shows a comparison of an ideal lowpass filter to a typical realizable lowpass filter.

6.2.2 FIR Lowpass Filters

The ideal lowpass filter has an infinitely long impulse response which decays slowly. In order to shorten the time response of the ideal lowpass, some form of truncation is used. This truncation process is the basis for many FIR filter design techniques. The truncation process (windowing) can not be made abrupt because the frequency response will be corrupted (the Gibbs Phenomenon). First, a common Hamming window will be used to truncate the ideal lowpass filter. Figure 6.3 shows the performance of a lowpass

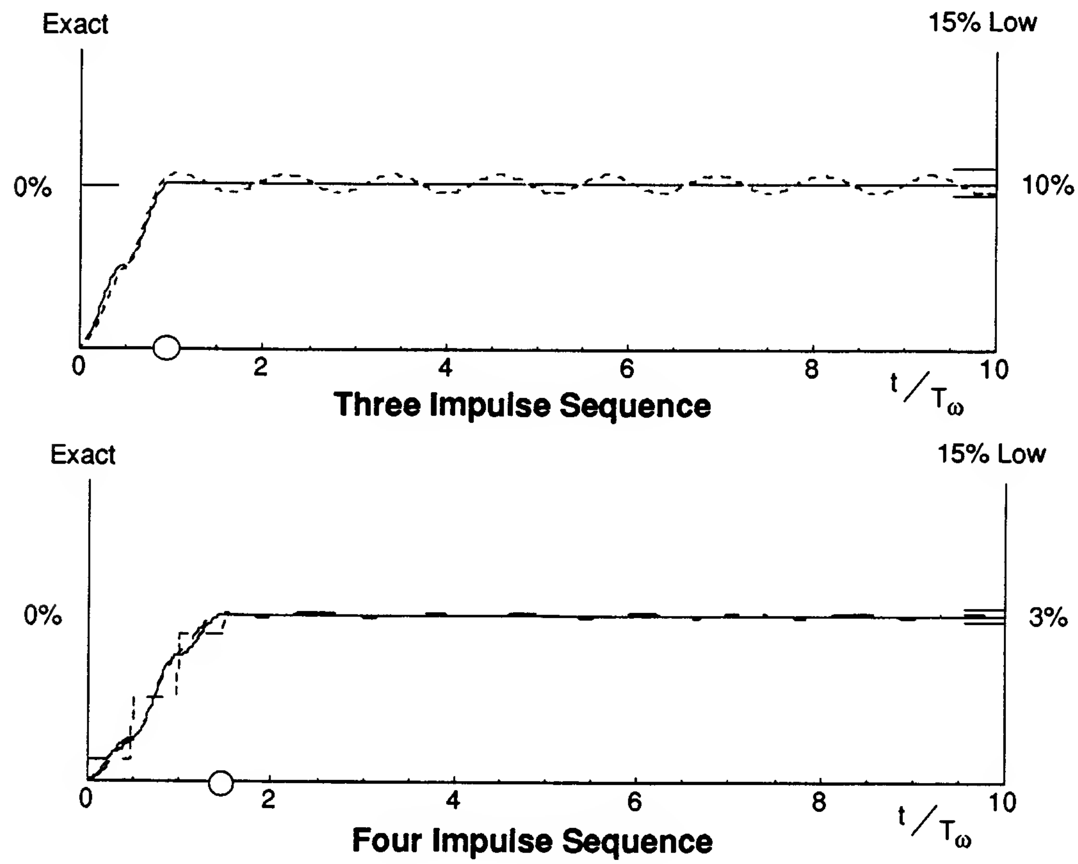


Figure 6.1: Comparison of responses of a simulated simple harmonic oscillator (SHO) to a unit step input passed to the system through various filters. Two responses are shown for each. The solid line is the response of an SHO with the exact natural frequency that was assumed during the design of the filters. The dashed line is the response of an SHO with a natural frequency that is 15% lower than the natural frequency used for the filter design. The residual vibration amplitude is marked on the axes. The time duration of each filter is marked with an open circle. T_ω is the vibrational period of the SHO system.

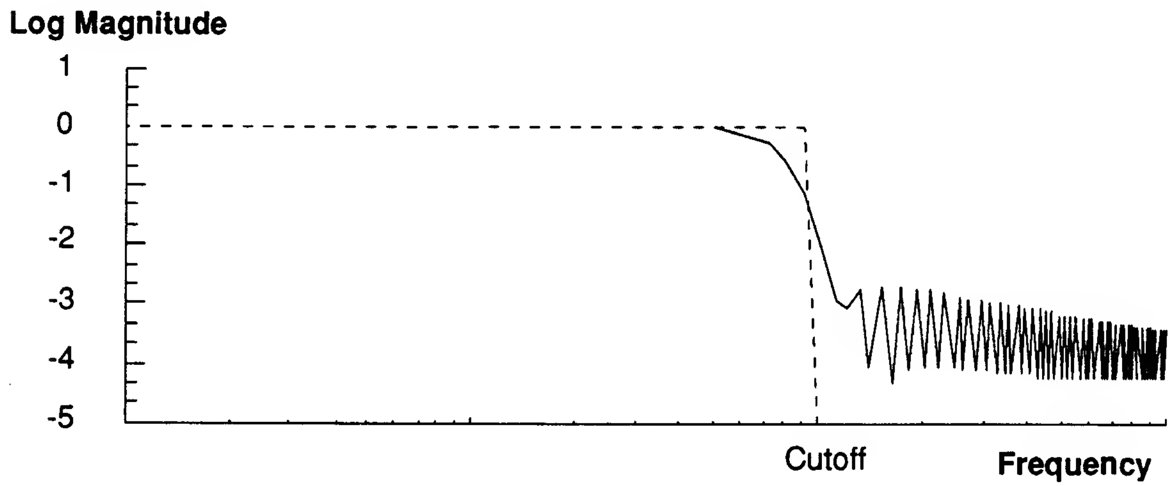


Figure 6.2: Ideal lowpass filter magnitude compared to a typical realizable lowpass filter. In this case the solid line is an FIR lowpass designed using a Hamming window.

filter designed using a Hamming window with a time duration of three system periods [105]. Figure 6.3 shows the performance for a longer filter (five system periods) designed with a Hamming window. These figures demonstrate that the filter length needs to be large compared to the period of system oscillation in order to achieve a reasonable response.

Figure 6.3 shows that the 5-second Hamming-windowed lowpass filter generates a step response with little vibration when the system is known. The drawback is that the time penalty is 5 seconds (or 60% longer than the three-impulse filter). An additional drawback is that there is little robustness. When the system natural frequency is lowered by 15%, a large, 30% residual vibration is incurred. The system response to the shorter, 3-second Hamming-windowed lowpass filter has considerably more vibration than either the 5 second version or the three-impulse sequence. Note that since both Hamming-window lowpass filters are true lowpass filters, if the system frequency is higher than expected, the vibration is always reduced. It is often suggested ..at this feature be exploited by setting the filter frequency significantly below the actual resonance. The

problem with this approach is that as the frequency of the filter is lowered, the time required for the filter sequence increases.

6.2.3 Parks-McClellan FIR Lowpass Filters

Next, FIR lowpass filters were designed using the Parks-McClellan-Rabiner technique [107]. The computation was performed with the original Remez exchange design program written by Parks and McClellan [107]. The highest possible filter length that double precision arithmetic allowed was used. Therefore, the filters in this paper are near the performance limit of Parks-McClellan designs and are computationally expensive. A 256 point FIR Filter was designed using the Parks-McClellan-Rabiner Algorithm. The filter was constrained to have a one system period time duration — the same time duration as the equivalent three-impulse shaping sequence. This filter was designed with a passband at 80% of the anticipated natural frequency, and a stopband at 95% of the anticipated natural frequency. Figure 6.3 demonstrates the effect of using this filter on the example system described above.

Note that the Parks-McClellan lowpass filter ends after 3 seconds. This was part of the design specifications. However, because the filter duration is short, the performance of this filter is poor. It performs better than the 3 second Hamming-windowed filter because the Parks-McClellan algorithm uses an optimality criterion to determine the filter coefficients while the Hamming technique does not. Even when the system is exact, a large, 14% residual vibration resulted.

6.2.4 Infinite Impulse Response Lowpass Filters

Oppenheim and Schafer [105] note that FIR filters provide greater "flexibility in the attainable filter response". However, infinite impulse response (IIR) filters are extremely common, therefore, they will be considered. IIR filters are limited because they must

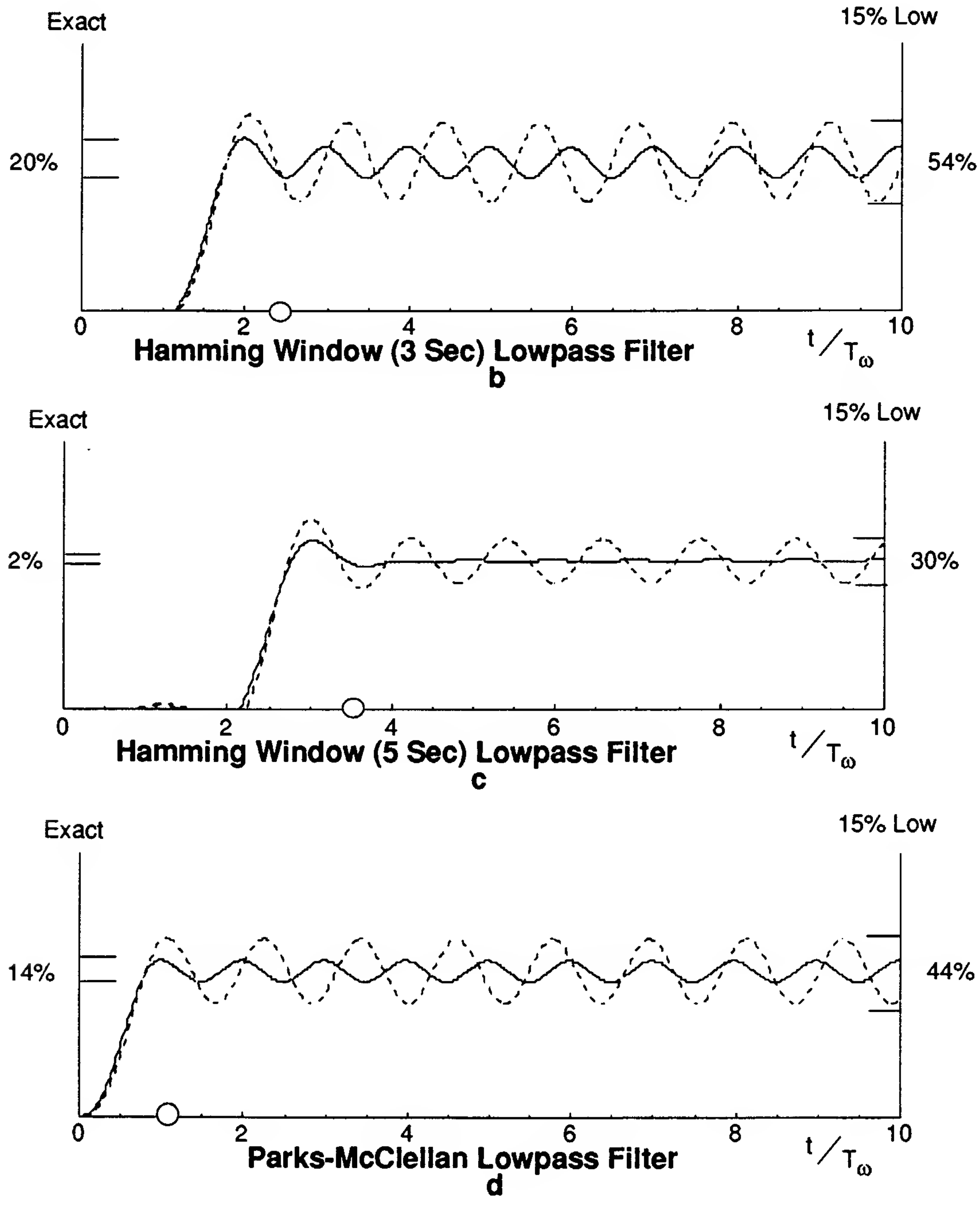


Figure 6.3: Comparison of responses of a simulated simple harmonic oscillator (SHO) to a unit step input passed to the system through various filters. Two responses are shown for each. The solid line is the response of an SHO with the exact natural frequency that was assumed during the design of the filters. The dashed line is the response of an SHO with a natural frequency that is 15% lower than the natural frequency used for the filter design. The residual vibration amplitude is marked on the axes. The time duration of each filter is marked with an open circle. T_ω is the vibrational period of the SHO system.

have poles and therefore, contain integration dynamics. Three common IIR filters will be examined — Butterworth, Chebyshev, and elliptic. The Butterworth and Chebyschev filters were designed with a passband at 70% of the anticipated natural frequency, and a stopband at 95% of the anticipated natural frequency. The Chebyschev filter had 2db ripple and 50db attenuation in the stopband. The elliptic filter was designed with a stopband at the anticipated natural frequency, and a passband at 70% of the anticipated natural frequency with 10 db ripple and 30db attenuation.

Figure 6.4 compare the system responses of the Butterworth, Chebyshev, and elliptic IIR filters. Large time and vibration penalties are incurred by using these filters. The time duration of the IIR filters was measured as the time at which the filter response has decayed to within 2% of its final value. These IIR Lowpass filters all have less residual vibration than the FIR lowpass filters. However, the time durations are considerably longer. The Butterworth filter had the shortest time duration of ≈ 5.8 seconds and a reasonably small residual vibration.

6.3 Notch Filters

6.3.1 Ideal Notch Filters

Ideal notch filters, like ideal lowpass filters, are not realizable. Figure 6.5 shows a comparison of an ideal notch filter to a typical realizable notch filter.

6.3.2 Hamming Window Notch Filter

A Hamming window can be applied to an ideal notch filter in the same manner as it was applied to the lowpass filter. The windowing process produces realizable filters with frequency responses close to that of the ideal filter. First, the ideal notch impulse response is determined. Next, the coefficients of the impulse response are multiplied

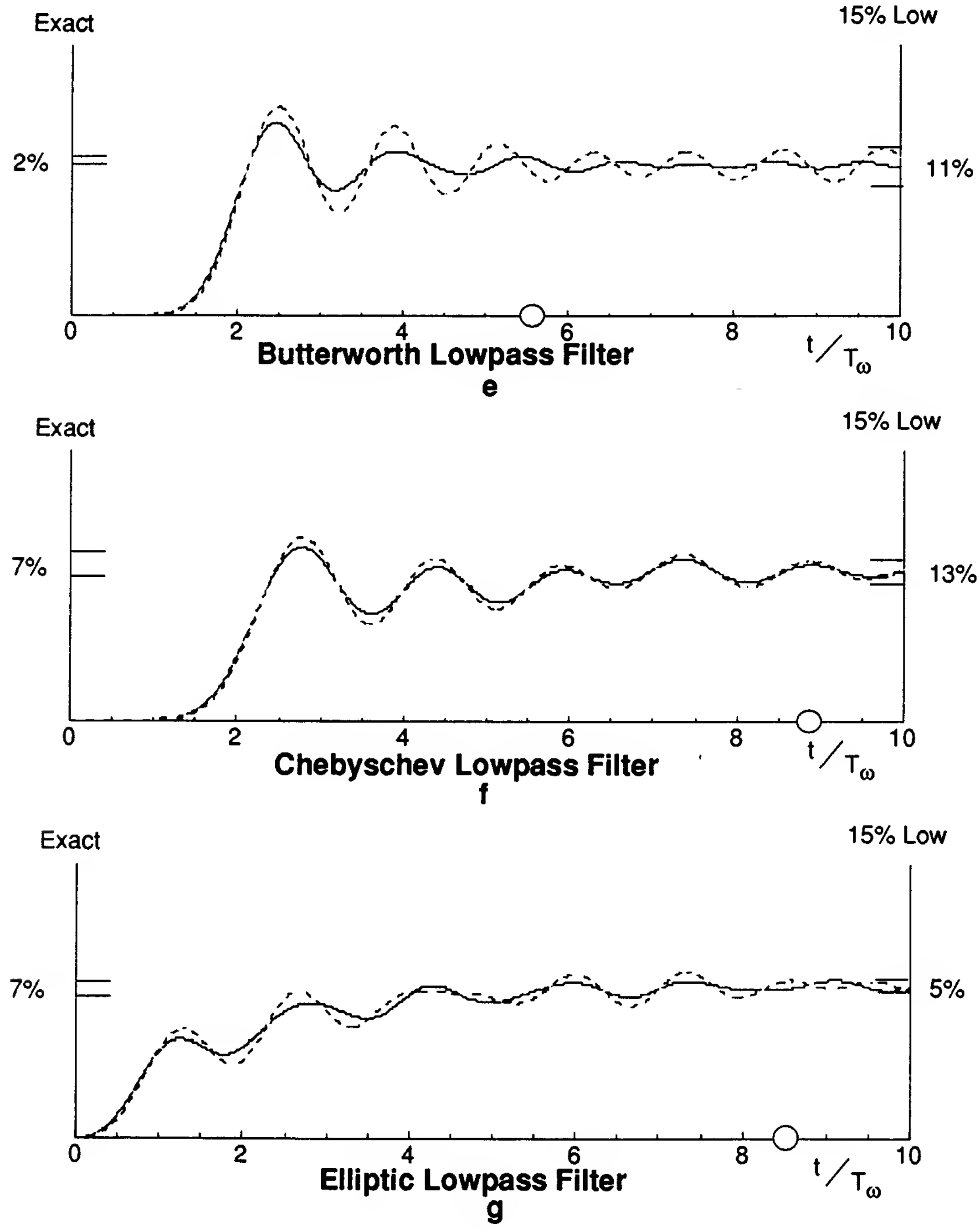


Figure 6.4: Comparison of responses of a simulated simple harmonic oscillator (SHO) to a unit step input passed to the system through various filters. Two responses are shown for each. The solid line is the response of an SHO with the exact natural frequency that was assumed during the design of the filters. The dashed line is the response of an SHO with a natural frequency that is 15% lower than the natural frequency used for the filter design. The residual vibration amplitude is marked on the axes. The time duration of each filter is marked with an open circle. T_ω is the vibrational period of the SHO system.

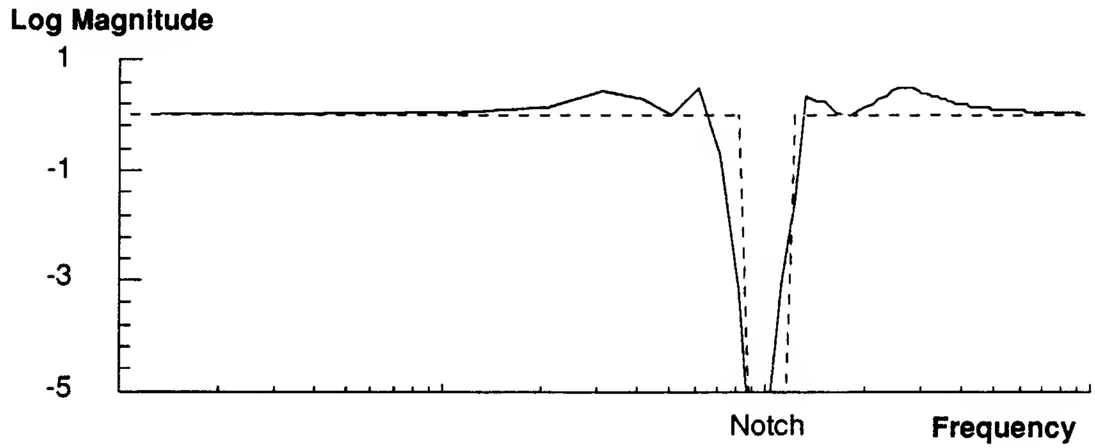


Figure 6.5: Ideal notch filter magnitude compared to a realizable notch filter. In this example the realizable filter is a Parks-McClellan notch filter.

by the values of the Hamming window. Figure 6.6 shows the effect of the applying a 5 period window to an ideal notch filter. The center of the notch is at the natural frequency of the system. The notch extends ±20% about the center frequency.

The hamming-windowed notch filter performed poorly. Even though it had a relatively long, 5 second duration, it resulted in a 32% residual vibration.

6.3.3 Parks-McClellan Notch Filters

FIR notch filters were also designed using the Parks-McClellan-Rabiner algorithm. A 256 point Parks-McClellan FIR filter is used for the comparisons. The filter was constrained to have the same time duration as the three-impulse shaping sequence (one system period). This filter was designed with a passband of ±20% about the anticipated natural frequency, and a stopband of ±5% about the anticipated natural frequency. Figure 6.6 shows the results of using this filter on a test system. The Parks-McClellan FIR design technique has no capabilities of modeling the decaying sinusoidal response of the system so this information can not be included in the filter derivation.

The Parks-McClellan notch filter performed the best of any of the filters shown

(except for the three- and four-impulse filters). The drawback of this filter is that a 256 point sequence was used. This filter is virtually impossible to implement in real time because 256 multiplies and additions must be performed at each time step. Chapter 5 implements a Parks-McClellan that is 5 points long (the most that could be run at a 1 Khz servo rate is ≈ 10). As the filter length is shortened, the performance of the filter degrades.

The reason that the Parks-McClellan algorithm does not produce the three- or four-impulse filters is because the Parks-McClellan algorithm spaces the points (the impulses) out at even intervals in time. The three- and four-impulse filters are successful because the timing of the impulses is carefully chosen (or in the case of digital systems, more pulses are included to compensate for the lack of precise timing). Additionally, the shaping sequences generated in chapters 3 and 4 for damped systems are not linear phase FIR filters while Parks-McClellan produces only linear phase filters.

6.3.4 Infinite Impulse Response Notch Filters

Figure 6.7 compares the system responses of the Butterworth, Chebyshev, and elliptic IIR notch filters. These filters were designed with a passband of $\pm 30\%$ about the anticipated natural frequency, and a stopband of $\pm 20\%$ about the anticipated natural frequency. Large time and vibration penalties are incurred by using these filters. They performed poorly in this application.

6.4 Summary

Table 6.1 compares some benchmarks for the simulations shown in the series of figures 6.1 through 6.7 . Three values are listed for each of the filtered systems which were commanded with a step input. The time duration of the filter is the number of system periods required for the output of the filter (which is the input to the system) to settle

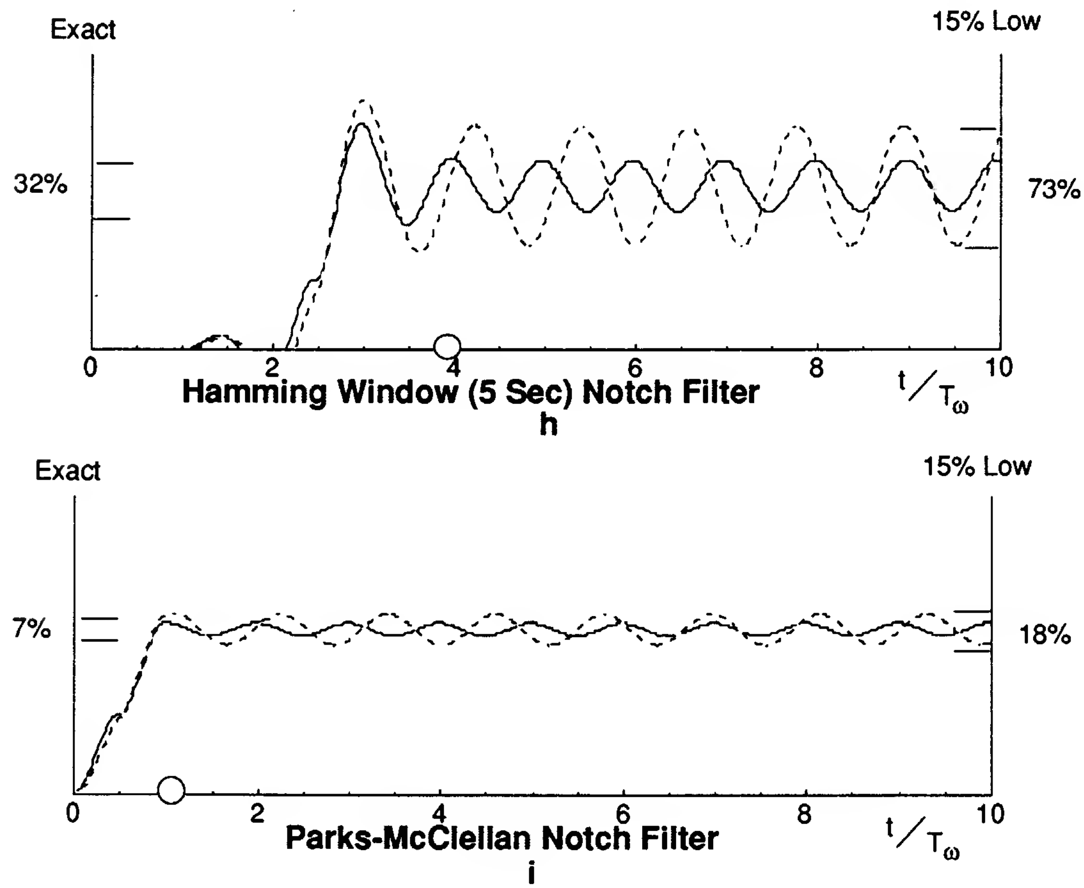


Figure 6.6: Comparison of responses of a simulated simple harmonic oscillator (SHO) to a unit step input passed to the system through various filters. Two responses are shown for each. The solid line is the response of an SHO with the exact natural frequency that was assumed during the design of the filters. The dashed line is the response of an SHO with a natural frequency that is 15% lower than the natural frequency used for the filter design. The residual vibration amplitude is marked on the axes. The time duration of each filter is marked with an open circle. T_ω is the vibrational period of the SHO system.

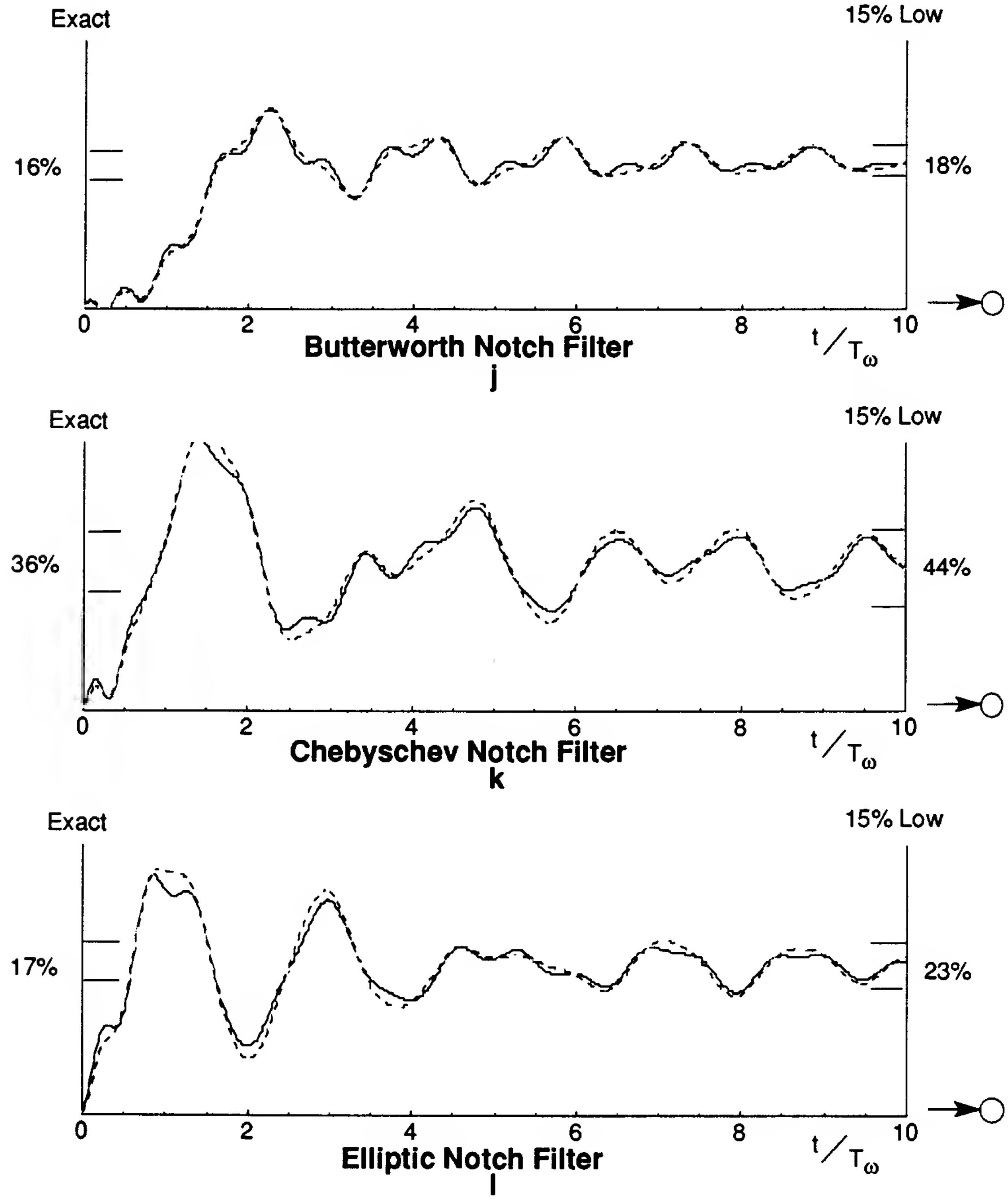


Figure 6.7: Comparison of responses of a simulated simple harmonic oscillator (SHO) to a unit step input passed to the system through various filters. Two responses are shown for each. The solid line is the response of an SHO with the exact natural frequency that was assumed during the design of the filters. The dashed line is the response of an SHO with a natural frequency that is 15% lower than the natural frequency used for the filter design. The residual vibration amplitude is marked on the axes. The time duration of each filter is marked with an open circle. T_ω is the vibrational period of the SHO system.

to within 2% of its steady state value. The vibration amplitude benchmarks are the amount of residual vibration that remains in the system expressed as a percentage of the move distance.

6.5 Conclusion

A comparison of the performance of a new shaping method to a variety of filters demonstrates that frequency domain filter design techniques are less effective for shaping vibration-reducing inputs to systems. The three and four impulse shaping sequence are shown to offer significant performance advantages in that moves require less time, and result in little or no residual vibration. In addition, the response is robust to variations in the system parameters.

Filter	Duration (cycles)	Vibration	
		Exact	15% Low
Three Impulse Sequence	1.00	0.0%	10%
Four Impulse Sequence	1.5	0.0%	3%
Hamming Window Lowpass (short)	2.46	20%	54%
Hamming Window Lowpass (Long	3.54	2.0%	30%
Parks-McClellan Lowpass	1.00	28%	57%
Butterworth Lowpass	5.61	2.0%	11%
Chebyschev Lowpass	8.90	7.2%	13%
Elliptical Lowpass	8.50	6.9%	5.4%
Hamming Window Notch	3.93	32%	73%
Parks-McClellan Notch	1.00	7.8%	18%
Butterworth Notch	>10.0	16%	18%
Chebyschev Notch	>10.0	36%	44%
Elliptical Notch	>10.0	17%	23%

Table 6.1: Summary of the comparison of the filters examined in this paper. The first number is the duration of the filter expressed in terms of the number of system cycles of vibration. The next two columns are the residual vibration expressed as a percentage of the move distance. The middle column is for a system with a natural frequency that is exactly the frequency that was used for the filter design. The last column is the residual vibration percentage for a system that has a natural frequency that is 15% lower than expected.

Literature Review

Chapter 7

7.1 Control of Flexible Systems

Several articles have been written on the topic of modeling and control of flexible systems. Balas [9] provides an excellent survey of this literature. Dubowsky [40] discusses the problems that need to be addressed in several fields of robotics. Nurre [103] surveyed the aerospace literature by examining the dynamics and control of large space structures. The next few sections present a moderately detailed discussion of the flexible system literature.

7.1.1 Modeling Flexible Systems

The first step toward controlling flexibles systems is developing the tools to generate good models. The models for flexible systems tend to be highly complex. Machines that are modeled by continuous structural elements that move through space lead to highly-coupled, nonlinear equations.

Many researchers have generated models of flexible systems and some have applied standard control approaches to the equations that resulted. Some examples are [1, 10,11,26,129,62,69,65,76,79,96,102,111,114,37,140]. The work of Truckenbrodt [132] is

noteworthy because of the experimental verification that is included. Bremer [25] and Barraco [13] compare different modeling approaches. Book [20,22,30] has presented a series of formulations for simplifying the generation of flexible models. Computer methods for symbolic equation generation have also been investigated [30,50]. Eppinger [44] shows what aspects of flexible plant models are important to consider.

7.1.2 Vibration Reduction

Robot vibration reduction research efforts can be classified into seven distinct approaches. Several of these techniques apply to existing robots while others involve redesign of the manipulator.

The first approach is to add structural damping or passive vibration absorbers to the manipulator. The goal of this is to make the structure damped to the extent that structural vibration can be ignored in the control strategy. Ideally, inputs would excite vibrations which would decay in magnitude fast enough so that the task of the robot would not be significantly delayed. Several researchers are investigating the use of piezoresistive films or elements which apply bending moments to the beam so as to resist the beam's motion [3,35,36,46,51,91,22,8,131]. Passive vibration absorption techniques have been extensively studied in the literature [60,91,106]. Damping is often difficult to add to systems. Active damping schemes have had limited success because of energy dissipation restrictions.

The second approach is to directly measure the absolute position of the endpoint of the manipulator. This position measurement can then be used in a conventional feedback loop to control out endpoint vibration such as in many of the works of Cannon [121,27,28,22,21,23,38,39,73,120,119]. This technique involves addressing the technical issues of noncollocated control. In addition, noise-free measurement of endpoint location without interfering with the robot's task is a difficult problem. Yurkovich [141,72]

and Chalhoub [31] have examined the use of endpoint acceleration feedback to damp vibrational response.

The third approach is to obtain a direct measurement of the modal response of the robot. This research direction requires addressing two major problems. The first is developing a method for measuring the vibration in a robot link. Some researchers have used distributed strain gages to measure the structural bending in each link [36, 8,110]. However, this approach must address noise issues. The second is developing a computational technique for distinguishing the different modes in the overall link deflection. Once the modal information is available, several approaches can be taken. For example, a control loop can be closed around each distinguished mode in order to add damping or to actively drive the axes in a manner which reduces vibration [10,11]. Many papers have examined the use of conventional feedback for the control of a measured vibrational state. Some examples are [9,22,28,29,46,52,67,70,75,78,81,100] Optimal feedback approaches have also been examined, for example, [24,52,82,94,115, 120,135,136,139]. Eldred [43] and Schaechter [116] control the shape of a beam using several actuators and feedback techniques. A good example of work which synthesises several approaches of feedback and feedforward is Pfeiffer [109].

The fourth approach is to use additional actuators to damp structural vibration. Some researchers have examined the use of lumped-actuators [141,110,83,142]. Others have examined distributed actuators [36,35,91,90] which are placed around the structure in order to damp vibrations throughout the system.

The fifth approach is to use a small positioner (micro-manipulator) at the endpoint of the large manipulator [57]. The small positioner would have a much higher bandwidth than the large robot. Therefore, the small manipulator would be able to compensate for small amplitude vibration in the large manipulator. This requires that the endpoint position or relative position be measurable so that it may be used as the set point for the micro positioner. In this configuration, the robot vibration would be treated as a

disturbance to the fine positioner. One difficult problem with this approach is that the compensating action of the fine positioner is necessarily at the exciting frequency of the manipulator, therefore, the micro positioner can excite the system, eventually driving the system unstable.

The sixth approach is to use a model of the system in order to generate feedforward command inputs which give the desired trajectory. Several techniques have been used for this approach. Some researchers use a linear model to generate trajectories [47]. Others have used an inversion of the nonlinear equations of the system [6]. A detailed examination of this literature is provided in the next section.

Many of the above have been used in an adaptive formulation in an effort to extend them to nonlinear and varying systems. Some examples of adaptive formulations specifically targeted for vibrating systems are [12,41,99,118,119,121,126,134,141].

7.2 Feedforward Command Shaping

7.2.1 Mechanical Systems

There are three aspects of commanding a machine's motion. The first is the problem of specifying a path for the robot. There is a great deal of literature on the topic of path planning and obstacle avoidance which generates desirable paths. Some examples can be found in [80,42]The second problem is that of specifying an exact trajectory that the robot can physically be able to follow. Often trajectories are requested that are beyond the limitations of the machine to reproduce. The third aspect is that of generating an input command that produces the desired trajectory, without oscillations, on the machine of interest. The third problem is the one which is addressed in this work and is referred to as "command shaping".

Command shaping involves preshaping either motor commands or setpoints so that

vibration is reduced. This aspect of control is often ignored because it is mistakenly considered to be useful only for open loop systems. However, if the input shaping accounts for the vibrational modes of the closed loop plant, then shaped input commands can be given to the closed loop plant as well. Thus, any of the preshaping techniques may be readily used as a closed loop technique [88,127].

The earliest form of command preshaping was the use of high-speed cam profiles as motion templates. These input shapes were generated so as to be continuous throughout one cycle (ie. the cycloidal cam profile). Their smoothness (continuous derivatives) reduces vibration by not putting high frequency inputs into the system [117]; however, these profiles have limited success.

Another early form of setpoint shaping for a one-mode system was the use of posicast control by O.J.M. Smith [127]. This technique involves breaking a step of a certain magnitude into two smaller steps, one of which is delayed in time. This results in a response with a reduced settling time. In effect, superposition of the responses leads to vibration cancellation. However, this is not generally used because of problems with robustness. The system that is to be commanded must have only one resonance, be known exactly, and be very linear for this technique to work. A version of Smith's posicast control was developed for moving nonlinear pendulum systems by Jones [59].

Optimal control approaches have been used to generate input profiles for commanding vibratory systems. Some examples are [4,17,21,45]. Junkins, Turner, Chun, and Juang have made considerable progress toward practical solutions of the optimal control formulation for flexible systems [63,61,33]. Dubowsky and Shiller [42] developed a method for generating trajectories that a machine can successfully follow. Typically, a penalty function is selected (for example integral squared error plus some control penalty). The resulting "optimal" trajectory is obtained in the form of the solution to the system equations (a model). This input is then given to the system.

Farrenkopf [45] and Swigert [130] demonstrated that velocity and torque shaping

can be implemented on systems which modally decompose into second order harmonic oscillators. They showed that inputs in the form of the solutions for the decoupled modes can be added so as not to excite vibration while moving the system. Their technique solves for parameters in a template function, therefore, inputs are limited to the form of the template. These parameters that define the control input are obtained by minimizing some cost function using an optimal formulation. The drawback of this approach is that the inputs are difficult to compute and they must be calculated for each move of the system.

Gupta [49], and Junkins and Turner [63] also included some frequency shaping terms in the optimal formulation. The derivative of the control input is included in the penalty function so that, as with cam profiles, the resulting functions are smooth. Several papers also address the closed loop "optimal" feedback gains which are used in conjunction with the "optimal" open-loop input. [63,61,33]

There are four drawbacks to these "optimal" approaches. First, computation is difficult. Each motion of the system requires recomputation of the control. Though the papers cited above have made major advances toward simplifying this step, it continues to be extremely difficult or impossible to solve for complex systems.

Second, the penalty function does not explicitly include a direct measure of the system vibration. Tracking error is used in the penalty function, therefore, all forms of error are essentially lumped together — the issue of vibration is not addressed directly. One side effect is that these approaches penalize residual vibration but allow the system to vibrate during the move. This leads to a lack of robustness under system uncertainties. Removing vibrational energy from a system is difficult especially under conditions of system uncertainty. Techniques that start a move, allowing the system to vibrate and then expect to remove that vibration later in the move lack robustness to slight parameter variations unless the vibration is measured. In addition, vibration is undesirable during a move as well as at the end.

Third, the solutions are limited to the domain of continuous functions. This is an arbitrary constraint which enables the solution of the problem. However, chapter 4 demonstrates that discontinuous functions exist which achieve excellent results. Fourth, the move time influences the effectiveness of optimal input strategies. Moves of different length will have different vibration excitation levels.

Another technique is based on the concept of the computed torque approach. The system is first modeled in detail. This model is then inverted — the desired output trajectory is specified and the required input needed to generate that trajectory is computed. For linear systems, this might involve dividing the frequency spectrum of the trajectory by the transfer function of the system, thus obtaining the frequency spectrum of the input. For nonlinear systems this technique involves inverting the equations for the model. [5,6,15,47,56,68,125]

Techniques that invert the plant have four problems. First, a trajectory must be selected. If the trajectory is impossible to follow, the plant inversion fails to give a usable result. Often a poor trajectory is selected to guarantee that the system can follow it, thus defeating the purpose of the input [15]. Second, a detailed model of the system is required. This is a difficult step for machines which are not extremely simple. Third, the plant inversion is not robust to variations in the system parameters because no robustness criterion has been included in the calculation. Fourth, this technique results in large move time penalties because the plant inversion process results in an acausal input (an input which exists before zero time). In order to use this input, it must be shifted in time thus increasing the move time.

Another approach to command shaping is the work of Meckl and Seering [84,85,86, 87,88]. They investigated several forms of feedforward command shaping. One approach they examined is the construction of input functions from either ramped sinusoids or versine functions. This approach involves adding up harmonics of one of these template functions. If all harmonics were included, the input would be a time optimal rectangular

(bang-bang) input function. The harmonics that have significant spectral energy at the natural frequencies of the system are discarded. The resulting input which is given to the system approaches the rectangular shape, but does not excite the resonances.

Aspinwall [7] proposed a similar approach which involves creating input functions by adding harmonics of a sine series. The coefficients of the series are chosen to minimize the frequency content of the input over a band of frequencies. Unlike Meckl, the coefficients were not selected to make the sine series approach a rectangular function, therefore, a large time penalty was incurred.

Wang, Hsia, and Wiederrich [137] proposed yet another approach for creating a command input that moves a flexible system while reducing the residual vibrations. They modeled the system in software and designed a PID controller for the plant that gave a desired response. They then examined the actual input that the controller gave to the software plant and used this for the real system. Next, they refined this input (the reference) with an iteration scheme that adds the error signal to the reference in order to get better tracking of the trajectory. This technique requires accurate modeling of the system and is not robust to parameter uncertainty. In addition, this technique has the implicit assumption that a good response can be achieved with a PID controller, In fact, systems with flexibility can not be given sufficient damping and a reasonable response time simply by adding a PID controller.

Often, a notch filter is proposed for input signal conditioning. This approach gives poor results for several reasons. First, a causal (real time) filter distorts the phase of the resulting signal. This effect is aggravated by lengthening the filter sequence of digital filters or by increasing the order of analog or recursive filters. Therefore, efforts to improve the frequency characteristics of a filter result in increased phase distortion. Also, penalties, such as filter ringing or long move times often result.

Singer and Seering [123] investigated an alternative approach of shaping a time optimal input by acausally filtering out the frequency components near the resonances. This

has an advantage over notch filtering in that phase distortion and ringing no longer pose a problem. The drawbacks of this approach [123] are the tradeoffs that must be made between fidelity in frequency and reduction of the move time.

Prucz [112] investigated a technique called pulse control. This technique involves pulsing a system at twice the natural frequency with impulse inputs. This technique is essentially equivalent to repeating the two-impulse input of Chapter 4.

7.2.2 Digital Signal Processing

The shaping techniques of chapter 4 resulted in various FIR filters. A brief summary of some related digital signal processing (DSP) work is proveded in this section. Some good general texts that discuss standard FIR filter techniques are Oppenheim and Schafer [105], and Stearns [128]. The DSP community has investigated many adaptive FIR filter formulations. It is likely that these will apply to adaptive versions of the techniques presented in this thesis. A summary of adaptive filters and equalizers is given in Mulgrew [93].

The field of adaptive antennae arrays is a spacial analog of the vibration-reducing problem. Fundamental work on adaptive arrays is presented by Monzingo and Miller [92]; and Hudson [58].

Speech enhancement involves strengthening particular frequencies that are present in a waveform. This field is essentially the inverse problem of vibration reduction. Lim [77] discusses the problem of speech enhancement.

Conclusion

Chapter 8

8.1 Summary

This document addressed many aspects of shaping commands so that residual system vibrations are reduced or eliminated. Several techniques for reducing vibrations were considered. First, the problem was examined in the frequency domain. The frequency content of input functions was examined. Several techniques for altering this frequency content were presented. Two main approaches were used. First, smooth gaussian shaped inputs were generated. Second, Some acausal filtering techniques were considered for producing template functions that do not have energy content at the resonances.

Next, some software models were developed for benchmark tests so that the various techniques could be appropriately compared. A new method for facilitating and automating equation generation for flexible systems was presented. This made the analysis and modeling of experimental systems easier. Then, Draper Laboratories' DRS Space Shuttle Remote Manipulator model was converted to work on the MIT Artificial Intelligence Laboratory's SUN Computers. This model would then serve as a test system for trying various vibration-reduction techniques and comparing their performances.

Subsequently, a new, time-domain technique was presented. This technique is based

on the concept of breaking an input into several components and delaying the components so that vibrations are canceled. Essentially, vibrating responses are generated out of phase so that the vibrations cancel. Next, this concept was altered so that it became feasible for real systems by adding robustness to the technique. Uncertainties in the natural frequencies of the system are shown to be accommodated by the modified technique.

Because excellent results were obtained by applying the new method on several different system simulations (including the DRS), the remainder of this document treated various extensions and applications of the technique. Some examples are shortening the sequences that are used for shaping inputs, shaping inputs to digital systems, multiple joint actuation, and straight line motion.

Next, The new technique was compared to a variety of conventional filter formulations. This section demonstrated that the new formulation is far better suited for eliminating residual vibration in mechanical systems. This chapter helps to clarify the different effect that the new, time domain constraint equations have on the resulting filter design.

The new technique for shaping inputs was then verified with hardware experimentation. An instrumented test facility was built so that it resembles "real" machines. A series of experimental results supported the theories and simulation results presented throughout the rest of this document.

Lastly, the implications of this technique concerning nonlinear systems was investigated. The limitations of the shaping were considered, and some modified, heuristic approaches were presented. For example, systems that quickly saturate in velocity can be sent shaped velocity inputs. This approach was demonstrated on the DRS model of the Space Shuttle manipulator.

8.2 Suggested Work

The success of the new shaping technique has inspired many ideas for future work. Several of the important directions will be outlined in this section.

The concept of robust signal shaping has only been used as a pre-filter for closed loop systems. The positive aspect of this configuration is that **any** controller can be used, the signal shaping is independent of the controller. An interesting research topic would be to examine closed-loop implementations of the shaping techniques. One major obstacle to reformulating this technique in a closed loop form is dealing with the transportation lag introduced by the prefilter. If a closed-loop formulation were developed, it would be useful, because not only would the commands result in vibrationless output, but the control corrections would also be made "vibrationless".

A second area of research is a more in depth examination of nonlinear systems. The robustness of the new technique, in its basic form, has been shown to have some applications to certain nonlinear systems. An interesting follow-up study is to consider other variations on the shaping approach that can accommodate large shifts in system frequency.

A third area of research is to try different constraint combinations in the linear programming formulation of chapter 3. One possibility is to consider a set of constraints that sample the vibration error expression over a range of frequencies and limits the error below a threshold. Under this scenario, the vibration error would never have to be exactly zero at any frequency. This relaxation of constraints could lead to shorter sequences.

An additional topic of study on nonlinear systems is to determine under what conditions each of the heuristic techniques should be applied. Several techniques were demonstrated, however, the particular strengths of one technique over another was not fully considered. For example, the concept of breaking the workspace of the machine

into several regions and using a different shaping sequence in each region was not fully explored. Techniques for "softening" the transition between sequences would be beneficial.

Chapter 4 has shown that the impulse sequences can be made arbitrarily short. However, a design strategy should be developed for taking full advantage of the sequence shortening. Actuator limitations and other performance constraints will have to be considered in determining the shortest sequence for a particular system. This work would yield some fundamental results regarding the limitations of how short the sequences can be made.

Another related topic that seems to be a rich research area is the development of a design strategy that capitalizes on the use of input shaping. The input shaping strategies that were presented can significantly reduce cycle times on machines that are plagued by structural resonances. However, machines that are currently designed to be extremely stiff may be able to take advantage of the new shaping techniques. By lightening the structure of a machine and/or boosting controller gains and actuator output, significant raw speed improvements can be achieved at the expense of structural vibration. By then using shaping techniques, the new, higher speed system can be commanded without the unwanted vibration.

Generating the Equations of Motion for Flexible Systems

Appendix A

A.1 Introduction

An approach is presented for deriving the equations of motion for systems with continuous flexible elements. This approach can be applied to any method for deriving the equations of motion; here both the Lagrangian and Kane [64] formulations are presented. In the Lagrangian method, this technique simplifies mathematical manipulation and requires no integrations by parts. For Kane's method, this technique eliminates the need to assume mode shapes for continuous members. For both the Lagrangian and Kane's methods, partial differential equations with boundary conditions are derived, yet the complexity of equation generation is reduced to that of a rigid-body problem. This technique was developed to facilitate computer generation of equations of motions for flexible systems.

Advances in computational power have enabled the analysis of increasingly complex dynamic systems. These systems often contain continuous members (such as beams) which must be modeled. The classical approaches for formulating the equations of

motion work well for systems of rigid bodies. However, they are difficult to apply to complex systems that contain both rigid and continuous elements. The modification, presented here, of the usual approach for formulating equations of motion enables the accommodation of continuous elements with the ease of lumped-parameter elements. The equations which result are the exact partial differential equations of the system of interest and may be solved using any of a variety of numerical algorithms. This modification can be applied to any formulation for deriving system equations of motion. It was developed to facilitate the generation by computer programs such as MACSYMA of differential equations describing flexible systems. However, it also provides advantages when the equations are generated by hand.

Several approaches are currently used to derive equations for systems with continuous members. The partial differential equations and boundary conditions can be derived by expressing the Lagrangian of the system and applying Hamilton's Principle (the variational approach)[34]. Hamilton's Principle requires the use of variational calculus, and integration by parts of the integral terms. The terms that correspond to the interior of continuous elements are manipulated in integral form, only to be converted in the last step to differential form. An alternative to this approach is to assume mode shapes for the continuous members and apply the rigid-body formulation of Hamilton's equations. Terms with mode shape integrals must then be manipulated.

Low and Vidyasagar [79] simplified this process by developing a formulation of Lagrange's equations for systems that exhibit flexibility. This formulation avoids the need for rederiving Lagrange's equations for each case.

Kane's method for generating the equations of motion requires that mode shapes be assumed in order to accommodate flexible (continuous) elements [64][66]. The limitations of generating only an assumed mode solution from either the variational or Kane technique are twofold. First, validity of the mode shape assumptions bound the accuracy of the solution. Second, the spatial integrals of these mode shapes must be calculated.

The use of mode-shape orthogonality aids with this calculation; however, the process is not simple.

For the technique described here, it is assumed that each continuous flexible memeber in a dynamic system is represented by one infinitesimal particle of mass. The resulting dynamical equations, although incomplete, contain all of the dynamic terms found in the equations of motion that result when flexible elements are treated as continuous. Subsequently, these incomplete equations of motion can be modified to include the dynamics of the entire continuous element.

A.2 Example

The problem that will be solved as an example is shown in figure A.1. It consists of a continuous bending beam attached to a cart with mass, M_a. An ideal spring (with constant, K_1) and damper (with value, b) are attached from the cart to ground at a distance, z. An additional spring (with constant, K_2) is attached to the other end of the beam. This spring's extension is specified by w measured to ground. The beam can be considered an aggregation of elemental particles denoted by m_b. Each particle is a distance x from the end of the beam attached at the cart. The motion of each particle is y_x measured to the neutral axis of the beam. In this problem, the equations are simpler if the coordinate y_x is chosen relative to ground (so that $y_0 = z$). However, the coordinate y_x is chosen to be relative to the mass, M_a, because this selection of coordinates produces an example which raises many of the issues which arise in more complicated problems.

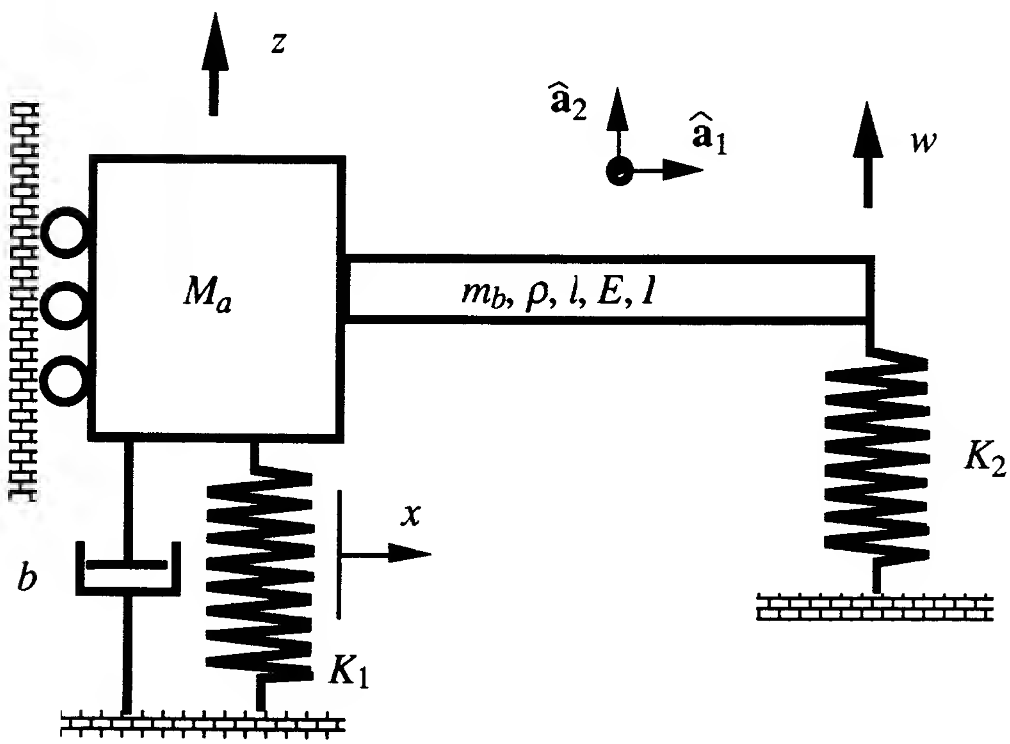


Figure A.1: Example problem

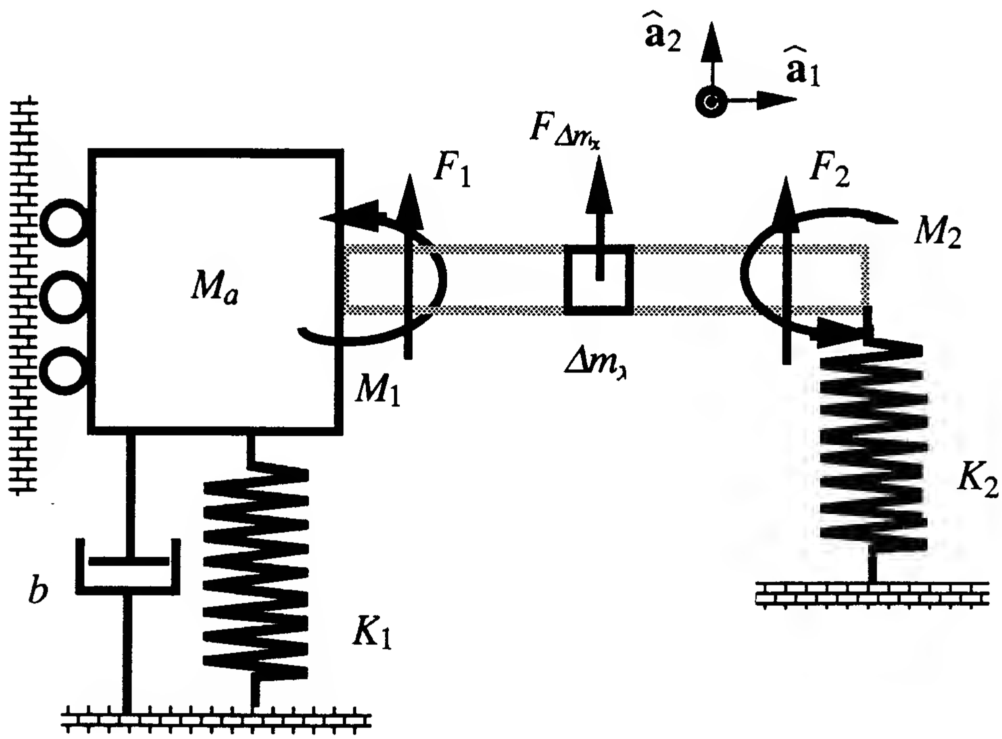


Figure A.2: Analogous example problem.

A.3 Generating a Rigid Particle Analogous System

Step 1 *Each continuous element in the problem is replaced by a single particle.* This particle is allowed to exhibit the motion prescribed by the continuum. For example, if a Bernoulli-Euler beam is being modeled, then the particle can move up or down perpendicular to the beam; if a Timoshenko beam model is used then the particle has two independent degrees of freedom — rotation and translation.

Figure A.2 demonstrates this step for the beam example. A Bernoulli-Euler beam is used, therefore, the beam particle with mass, Δm_x, can move up or down according to coordinate y_x. The particle can rotate but its rotation is not an independent degree-of-freedom since the Bernoulli-Euler model couples deflection and rotation. This coupling is due to the fact that shear deformation and rotational inertia are excluded from the

model.

Step 2 *Assign degrees of freedom to the rest of system.* The motion constraints are those of the system external to the continuum. The continuum does not impose motion constraints — it supplies only interface forces which are determined in the next step.

In the example, Mass M_a has a yet unknown force and moment placed on it by the beam particles. It can only translate up or down because of the rolling constraint. Therefore, it is assigned a coordinate, z.

A second degree of freedom is placed at the other end of the continuum on the spring. It also is subjected to a yet unknown force and a moment because it is connected to the continuum. Here the geometry of the system allows this point of the system to both rotate and translate. Therefore, it is assigned two degrees-of-freedom, w and θ_e. Note that it is **not** subject to the geometric constraints of a continuum particle.

Step 3 *Place forces and moments applyed* **by** *the continuum* **on** *the rest of the system.* This is to enforce the connection of the continuous particle to the rest of the system. Forces or moments that can not do any work are discarded. Care must be taken to correctly determine the signs of these forces and moments. Table A.1 gives a consistent summary of these values for standard continuous elements. The values for more unusual elements can be determined from basic principles. This calculation is not extra work. In Kane's method this calculation is necessary since the method relies on force expressions to derive the equations. In the Lagrangian technique, an unusual element requires the calculation of new potential and kinetic energy functions. The derivation of these energy expressions requires knowledge of the forces within the continuum. The most important aspect of this step is to generate a consistent sign convention.

Once again, the beam example is considered. The mass, M_a is connected to the continuum by including the force that the continuum exerts on it, $\mathbf{F}_1$. The moment

that the beam exerts on this mass can be ignored because the mass can not rotate, thus this moment can do no work. Likewise, the spring communicates with the beam continuum through a force, $\mathbf{F}_2$ and moment, $\mathbf{M}_\tau$ because this location can have both a rotational and translational degree-of-freedom. Note that the equal but opposite forces and moments to $\mathbf{F}_1$, $\mathbf{F}_2$, $\mathbf{M}_\tau$ are not explicitly added because they will be automatically included in the two forces and moments on the neighboring beam particle (represented as Δm_x).

The values of these forces and moments for the example are determined as follows:

$$F_1 = -\frac{\partial}{\partial x}\left[EI\frac{\partial^2 y_x}{\partial x^2}\right]\Bigg|_{x=0} \tag{A.1}$$

$$F_2 = -\frac{\partial}{\partial x}\left[EI\frac{\partial^2 y_x}{\partial x^2}\right]\Bigg|_{x=L} \tag{A.2}$$

$$M_\tau = EI\,\frac{\partial^2 y_x}{\partial x^2}\Bigg|_{x=L} \tag{A.3}$$

Once these steps have been completed, the problem has been fully transformed into a new, analogous, rigid-particle system. The equations for this new system are now solved using any standard approach. The next two sections present the solution using the Lagrangian and Kane approaches respectively.

A.4 Lagrangian Approach

A.4.1 The New Procedure

The common expression for Lagrange's equations:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\xi}_j}\right) - \frac{\partial \mathcal{L}}{\partial \xi_j} = \Xi_j \qquad j = 1, 2, \ldots, n \tag{A.4}$$

is only valid for Lagrangians of the form,

$$\mathcal{L}(\xi_1, \ldots, \xi_n; \frac{d\xi_1}{dt}, \ldots, \frac{d\xi_n}{dt}; t) \quad .$$

It is important to note that in the case when a single rigid particle is used in the place of the continuous element, the potential energy terms are no longer functions of the ξ_i. These terms in V become functions of the derivatives of the ξ_i, $\frac{\partial \xi_i}{\partial x}$ and $\frac{\partial^2 \xi_i}{\partial x^2}$. The Lagrangian in this case becomes:

$$\mathcal{L}(\xi_1, \ldots, \xi_n; \frac{\partial \xi_1}{\partial t}, \ldots, \frac{\partial \xi_n}{\partial t}; \frac{\partial^2 \xi_1}{\partial x^2}, \ldots, \frac{\partial^2 \xi_n}{\partial x^2}; \frac{\partial \xi_1}{\partial x}, \ldots, \frac{\partial \xi_n}{\partial x}; t)$$

In order to apply the calculus of variations to Lagrangians of this new form, Lagrange's equations must be generalized to:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\xi}_j}\right) - \frac{\partial \mathcal{L}}{\partial \xi_j} + \frac{\partial}{\partial x}\left(\frac{\partial \mathcal{L}}{\partial (\frac{\partial \xi_j}{\partial x})}\right) - \frac{\partial^2}{\partial x^2}\left(\frac{\partial \mathcal{L}}{\partial (\frac{\partial^2 \xi_j}{\partial x^2})}\right) = \Xi_j \qquad j = 1, 2, \ldots, n \qquad \text{(A.5)}$$

The first two terms are the same as Lagrange's equations. The last two terms are added in order to handle the derivative terms that now appear in the Lagrangian. These equations are essentially the same as those derived in [79][74].

Step 4 *Use the new form of Lagrange's equations to find the equations of motion for the analogous rigid body system.* First, the values for the kinetic coenergy, T^*, and the potential energy, V are determined. This step is essentially the same as that of the standard variational approach. The kinetic energy term is just the sum of one half of the particles' mass multiplied by their respective velocities squared. The potential terms are the same as in the rigid body case with the addition of terms for the continuous elements. Table A.1 gives these terms for various continuums. Note that the potential terms for the continuum all are multiplied by Δx. This is important because these potential energy terms are, in fact, densities. They are integrated in x when using the standard formulation of Hamilton's Principle. In this technique, they are not integrated and are applied at a particle, therefore, they must be multiplied by Δx.

After the Lagrangian is formed, the variational approach is performed in the same manner as the standard formulation yeilding n equations of motion for the analogous

system. It is important to note that use of the more general form of Lagrange's equations (A.5) requires no more computation than the equations (A.4) applied to a rigid-body system. Most of the terms in the Lagrangian do not contain the $\frac{\partial \xi_j}{\partial x}$ or the $\frac{\partial^2 \xi_j}{\partial x^2}$, and the terms that do contain them are replacing terms that would have contained ξ_j if the system truly were a rigid-body system. Section A.6 presents the process for converting the resulting equations into the equations for the continuous system.

A.4.2 Example Problem; Variational Approach

For our example, the rigid-body equivalent system is shown in figure A.2 for the elemental particle at location, x.

Step 4.a The independent coordinates are chosen as:

$$\xi_1 = z \tag{A.6}$$

$$\xi_2 = y_x \tag{A.7}$$

$$\xi_3 = w \tag{A.8}$$

$$\xi_4 = \theta_e \tag{A.9}$$

Step 4.b The kinetic coenergy, potential energy, and Lagrangian are expressed as:

$$T^* = \frac{1}{2}\Delta m_x \left(\frac{\partial y_x}{\partial t} + \frac{dz}{dt}\right)^2 + \frac{1}{2}M_a \left(\frac{dz}{dt}\right)^2 \tag{A.10}$$

$$V = \frac{1}{2}EI\left(\frac{\partial^2 y_x}{\partial x^2}\right)^2 \Delta x + \frac{1}{2}K_1 z^2 + \frac{1}{2}K_2 w^2 \tag{A.11}$$

$$\mathcal{L} = T^* - V \tag{A.12}$$

Note that V is now a function of $\frac{\partial^2 y_x}{\partial x^2}$ (or $\frac{\partial^2 \xi_2}{\partial x^2}$).

Step 4.c Using the principle of virtual work, the generalized forces, Ξ_i, associated with each degree-of-freedom are determined:

$$\Xi_1 = F_1 - b\frac{dz}{dt} \tag{A.13}$$

$$\Xi_2 = q(x)\Delta x \tag{A.14}$$

$$\Xi_3 = -F_2 \tag{A.15}$$

$$\Xi_4 = M_\tau \tag{A.16}$$

Step 4.d Apply the new form of Lagrange's equations (A.5) for $\xi_1 = z$:

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{z}}\right) &= \frac{\partial}{\partial t}\left[\Delta m_x \left(\frac{\partial y_x}{\partial t} + \frac{dz}{dt}\right) + M_a \frac{dz}{dt}\right] \\
-\frac{\partial \mathcal{L}}{\partial z} &= K_1 z \\
\frac{\partial}{\partial x}\left(\frac{\partial \mathcal{L}}{\partial \frac{\partial z}{\partial x}}\right) &= 0 \\
-\frac{\partial^2}{\partial x^2}\left(\frac{\partial \mathcal{L}}{\partial \frac{\partial^2 z}{\partial x^2}}\right) &= 0 \quad ,
\end{aligned} \tag{A.17}$$

for $\xi_2 = y_x$:

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{y_x}}\right) &= \frac{\partial}{\partial t}\left[\Delta m_x \left(\frac{\partial y_x}{\partial t} + \frac{dz}{dt}\right)\right] \\
-\frac{\partial \mathcal{L}}{\partial y_x} &= 0 \\
\frac{\partial}{\partial x}\left(\frac{\partial \mathcal{L}}{\partial (\frac{\partial y_x}{\partial x})}\right)\Delta x &= 0 \\
-\frac{\partial^2}{\partial x^2}\left(\frac{\partial \mathcal{L}}{\partial (\frac{\partial^2 y_x}{\partial x^2})}\right) &= \frac{\partial^2}{\partial x^2}\left[EI\frac{\partial^2 y_x}{\partial x^2}\right] \quad ,
\end{aligned} \tag{A.18}$$

for $\xi_3 = w$:

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{w}}\right) &= 0 \\
-\frac{\partial \mathcal{L}}{\partial w} &= K_2 w \\
\frac{\partial}{\partial x}\left(\frac{\partial \mathcal{L}}{\partial (\frac{\partial w}{\partial x})}\right) &= 0 \\
-\frac{\partial^2}{\partial x^2}\left(\frac{\partial \mathcal{L}}{\partial (\frac{\partial^2 w}{\partial x^2})}\right) &= 0 \quad ,
\end{aligned} \tag{A.19}$$

and for $\xi_4 = \theta_e$:

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_e}\right) &= 0 \\ -\frac{\partial \mathcal{L}}{\partial \theta_e} &= 0 \\ \frac{\partial}{\partial x}\left(\frac{\partial \mathcal{L}}{\partial (\frac{\partial \theta_e}{\partial x})}\right) &= 0 \\ -\frac{\partial^2}{\partial x^2}\left(\frac{\partial \mathcal{L}}{\partial (\frac{\partial^2 \theta_e}{\partial x^2})}\right) &= 0 \quad . \end{aligned} \tag{A.20}$$

Step 4.e Compiling the equations of motion gives:

$$\Delta m_x\left(\frac{\partial^2 y_x}{\partial t^2} + \frac{d^2 z}{dt^2}\right) + M_a\frac{d^2 z}{dt^2} + K_1 z = -\frac{\partial}{\partial x}\left[EI\frac{\partial^2 y_x}{\partial x^2}\right]\Big|_{x=0} - b\frac{dz}{dt} \tag{A.21}$$

$$\Delta m_x\left(\frac{\partial^2 y_x}{\partial t^2} + \frac{d^2 z}{dt^2}\right) + \frac{\partial^2}{\partial x^2}\left[EI\frac{\partial^2 y_x}{\partial x^2}\right]\Delta x = q(x)\Delta x \tag{A.22}$$

$$K_2 w = \frac{\partial}{\partial x}\left[EI\frac{\partial^2 y_x}{\partial x^2}\right]\Big|_{x=L} \tag{A.23}$$

$$\frac{\partial^2 y_x}{\partial x^2}\Big|_{x=L} = 0 \tag{A.24}$$

The four equations (A.21) – (A.24) are altered according to Section A.6 to yield the system partial differential equations.

A.5 Kane's Method Approach

The extension presented in this paper is particularly useful in Kane's method for three reasons. First, Kane's method, until now, could not be used to generate partial differential equations for continuous systems (as far as we know). Only assumed-mode approximate differential equations were previously possible (Kane demonstrates only this approach [65]). Second, Kane's method can handle more complex problems than the Lagrangian technique. Third, Kane's method is not altered by this extension – the

same steps are performed and the resulting equations are modified to account for the system flexibility. The system is specified as a rigid body and rigid particle system as presented in section A.3. Kane's method is applied to this system as if it were a rigid body system and the equations of motion are generated. Section A.6 describes the method for transforming these lumped-parameter equations into the correct, continuous form.

A.5.1 Example Problem; Kane's Method

Step 4.a The particle model (figure A.1) from section A.4.2 is considered. The generalized speeds are selected as:

$$\begin{aligned}
u_1 &\triangleq \tfrac{dz}{dt} = \dot{z} \\
u_2 &\triangleq \tfrac{dy_x}{dt} = \dot{y_x} \\
u_3 &\triangleq \tfrac{dw}{dt} = \dot{w} \\
u_4 &\triangleq \tfrac{d\theta_e}{dt} = \dot{\theta}_e
\end{aligned}$$

Step 4.b Next, the velocities of the masses and the points of external force application are specified:

$$\begin{aligned}
{}^I\mathbf{V}^{M_a} = {}^I\mathbf{V}^{\mathbf{F}_{K_1}} = {}^I\mathbf{V}^{\mathbf{F}_1} = {}^I\mathbf{V}^{\mathbf{F}_b} &= \dot{z}\mathbf{a}_2 \\
{}^I\mathbf{V}^{\Delta m_x} = {}^I\mathbf{V}^{\mathbf{F}_{\Delta m_x}} &= (\dot{z} + \dot{y_x})\,\mathbf{a}_2 \\
{}^I\mathbf{V}^{\mathbf{F}_{K_2}} = {}^I\mathbf{V}^{\mathbf{F}_2} &= \dot{w}\mathbf{a}_2 \\
{}^I\mathbf{V}^{\mathbf{F}_r} &= \dot{\theta}_e\mathbf{a}_2
\end{aligned}$$

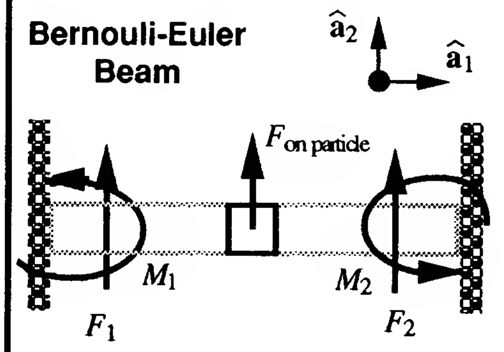
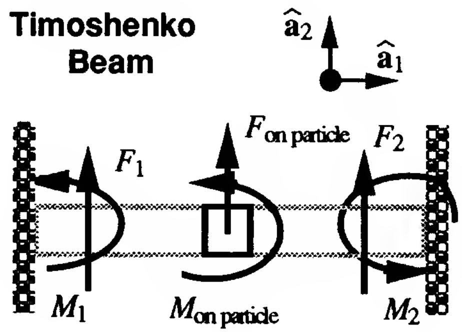
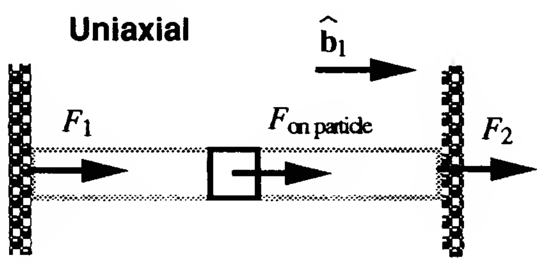
Step 4.c , the active forces are specified (See table A.1):

$$\mathbf{F}_{K_1} = -K_1 z\mathbf{a}_2 \tag{A.25}$$

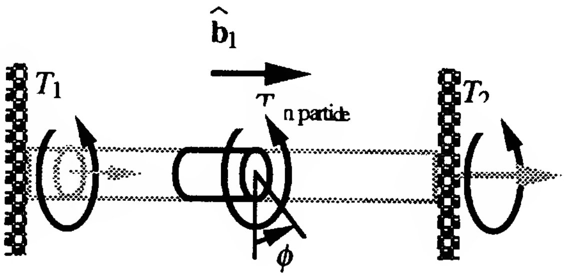
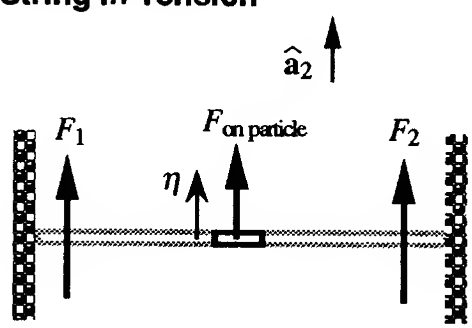
$$\mathbf{F}_{K_2} = -K_2 w\mathbf{a}_2 \tag{A.26}$$

$$\mathbf{F}_1 = -\frac{\partial}{\partial x}\left[EI\frac{\partial^2 y_x}{\partial x^2}\right]\Bigg|_{x=0}\mathbf{a}_2 \tag{A.27}$$

Continuum Sign Conventions

Continuum	Boundary Forces.	Particle Forces [1]	Particle Energy Terms
Bernouli-Euler Beam 	$F_1 = -\frac{\partial}{\partial x}\left(EI\frac{\partial^2 y}{\partial x^2}\right)\Big\vert_{x=0}\hat{\mathbf{a}}_2$ $F_2 = +\frac{\partial}{\partial x}\left(EI\frac{\partial^2 y}{\partial x^2}\right)\Big\vert_{x=l}\hat{\mathbf{a}}_2$ $M_1 = EI\frac{\partial^2 y}{\partial x^2}\Big\vert_{x=0}\hat{\mathbf{a}}_3$ $M_2 = -EI\frac{\partial^2 y}{\partial x^2}\Big\vert_{x=l}\hat{\mathbf{a}}_3$	$F_{\text{on particle}} = -\frac{\partial^2}{\partial x^2}\left[EI\frac{\partial^2 y}{\partial x^2}\right]\Big\vert_{x=0}\Delta x\hat{\mathbf{a}}_2$	$T^* = \frac{1}{2}\Delta m\left(\frac{\partial y}{\partial t}\right)^2$ $V = \frac{1}{2}EI\left(\frac{\partial^2 y}{\partial x^2}\right)^2\Delta x$
Timoshenko Beam 	$F_1 = \kappa GA\left[\frac{\partial \eta}{\partial x} - \phi\right]\Big\vert_{x=0}\hat{\mathbf{a}}_2$ $F_2 = -\kappa GA\left[\frac{\partial \eta}{\partial x} - \phi\right]\Big\vert_{x=l}\hat{\mathbf{a}}_2$ $M_1 = EI\frac{\partial \phi}{\partial x}\Big\vert_{x=0}\hat{\mathbf{a}}_3$ $M_2 = -EI\frac{\partial \phi}{\partial x}\Big\vert_{x=l}\hat{\mathbf{a}}_3$	$F_{\text{on Particle}} = \frac{\partial}{\partial x}\left[\kappa GA\left(\frac{\partial \eta}{\partial x} - \phi\right)\right]\Delta x\hat{\mathbf{a}}_2$ $M_{\text{on Particle}} = \frac{\partial}{\partial x}\left[EI\frac{\partial \phi}{\partial x}\right]\Delta x\hat{\mathbf{a}}_3$ $+ \kappa GA\left(\frac{\partial \eta}{\partial x} - \phi\right)\Delta x\hat{\mathbf{a}}_3$	$T^* = \frac{1}{2}\Delta m\left(\frac{\partial \eta}{\partial t}\right)^2 + \frac{1}{2}\Delta I\left(\frac{\partial \phi}{\partial t}\right)^2$ $V = \frac{1}{2}EI\left(\frac{\partial \phi}{\partial x}\right)^2\Delta x$ $+\frac{1}{2}\kappa GA\left(\frac{\partial \eta}{\partial x} - \phi\right)^2\Delta x$
Uniaxial 	$F_1 = EA\frac{\partial \xi}{\partial x}\Big\vert_{x=0}\hat{\mathbf{b}}_1$ $F_2 = -EA\frac{\partial \xi}{\partial x}\Big\vert_{x=l}\hat{\mathbf{b}}_1$	$F_{\text{on particle}} = \frac{\partial}{\partial x}\left(EA\frac{\partial \xi}{\partial x}\right)\Delta x\hat{\mathbf{b}}_1$ [1] Forces acting **on** the particle	$T^* = \frac{1}{2}\Delta m\left(\frac{\partial \xi}{\partial t}\right)^2$ $V = \frac{1}{2}EA\left(\frac{\partial \xi}{\partial x}\right)^2\Delta x$

Continuum Sign Conventions

Continuum	Interface Forces	Particle Forces	Energy Terms
Torsion 	$T_1 = GI_p \frac{\partial \phi}{\partial x}\Big\vert_{x=0} \hat{\mathbf{b}}_1$ $T_2 = -GI_p \frac{\partial \phi}{\partial x}\Big\vert_{x=l} \hat{\mathbf{b}}_1$	$T_{\text{on particle}} = \frac{\partial}{\partial x}\left(GI_p \frac{\partial \phi}{\partial x}\right)\Delta x \hat{\mathbf{b}}_1$	$T^* = \frac{1}{2}\Delta I\left(\frac{\partial \phi}{\partial t}\right)^2$ $V = \frac{1}{2}GI_p\left(\frac{\partial \phi}{\partial x}\right)^2 \Delta x$
String in Tension 	$F_1 = P\frac{\partial \eta}{\partial x}\Big\vert_{x=0} \hat{\mathbf{a}}_2$ $F_2 = -P\frac{\partial \eta}{\partial x}\Big\vert_{x=l} \hat{\mathbf{a}}_2$	$F_{\text{on particle}} = \frac{\partial}{\partial x}\left(P\frac{\partial \eta}{\partial x}\right)\Delta x \hat{\mathbf{a}}_2$	$T^* = \frac{1}{2}\Delta m\left(\frac{\partial \eta}{\partial t}\right)^2$ $V = \frac{1}{2}P\left(\frac{\partial \eta}{\partial x}\right)^2 \Delta x$

Definitions:

l = Length of Continuum x = Dimension along Continuum

E = Young's Modulus
G = Shear Modulus
A = Cross Section Area
P = String Tension

I_p = Torsional Constant
I = Area Moment of Inertia
ΔI = Particle Area Moment of Inertia
Δm = Mass of Elemental Particle

κ = Shear Coefficient

Shape:	κ
Rectangle	= .83
Circle	= .85
Typical I- Beam	= .95

η = String, Vertical
ξ = Uniaxial Stretch
ϕ = Rotation of Beam Particle
y = Beam, Verticle

Ref [18, 34, 64]

$$\mathbf{F}_2 = \frac{\partial}{\partial x}\left[EI\frac{\partial^2 y_x}{\partial x^2}\right]\Big|_{x=L}\mathbf{a}_2 \tag{A.28}$$

$$\mathbf{F}_{\Delta m_x} = -\frac{\partial^2}{\partial x^2}\left[EI\frac{\partial^2 y_x}{\partial x^2}\right]\Delta x\mathbf{a}_2 \tag{A.29}$$

$$\mathbf{F}_b = -b\dot{z}\mathbf{a}_2 \tag{A.30}$$

Step 4.d A partial velocity table is created:

	${}^I\mathbf{V}^{M_a}$, ${}^I\mathbf{V}^{\mathbf{F}_{K_1}}$, ${}^I\mathbf{V}^{\mathbf{F}_1}$, ${}^I\mathbf{V}^{\mathbf{F}_b}$	${}^I\mathbf{V}^{\Delta m_x}$, ${}^I\mathbf{V}^{\mathbf{F}_{\Delta m_x}}$	${}^I\mathbf{V}^{\mathbf{F}_{K_2}}$, ${}^I\mathbf{V}^{\mathbf{F}_2}$	${}^I\mathbf{V}^{\mathbf{F}_r}$
$\dot{z}$	$\mathbf{a}_2$	$\mathbf{a}_2$	0	0
$\dot{y}_x$	0	$\mathbf{a}_2$	0	0
$\dot{w}$	0	0	$\mathbf{a}_2$	0
$\dot{\theta}_e$	0	0	0	$\mathbf{a}_3$

Step 4.e The mass accelerations are calculated from the mass velocities:

$${}^I\mathbf{a}^{M_a} = \ddot{z}\mathbf{a}_2 \tag{A.31}$$

$${}^I\mathbf{a}^{\Delta m_x} = (\ddot{z} + \ddot{y}_x)\,\mathbf{a}_2 \tag{A.32}$$

Step 4.f Then, the inertia forces are determined for each of the m bodies from:

$$\mathbf{F}^{*B_j} = -M_{B_j}\,{}^I\mathbf{a}^{B_j} \qquad (j = 1, \ldots, m) \tag{A.33}$$

Step 4.g These inertia forces are used to form the generalized inertia forces from:

$$F_r^* = \sum_{j=1}^{m} {}^I\mathbf{v}_r{}^{B_j} \cdot \mathbf{F}^{*B_j} + {}^I\boldsymbol{\omega}_r{}^{B_j} \cdot \mathbf{T}^{B_j} \qquad (r = 1, \ldots, n) \tag{A.34}$$

For this example these are:

$$F_1^* = -M_a\ddot{z} - \Delta m_x(\ddot{z} + \ddot{y}_x) \tag{A.35}$$

$$F_2^* = -\Delta m_x(\ddot{z} + \ddot{y}_x) \tag{A.36}$$

$$F_3^* = 0 \tag{A.37}$$

$$F_4^* = 0 \tag{A.38}$$

Step 4.h Next, the generalized active forces are determined from the p forces, P_j by

$$F_r = \sum_{j=1}^{p} {}^I\mathbf{v}_r{}^{P_j} \cdot \mathbf{F}^{P_j} + {}^I\omega_r{}^{P_j} \cdot \mathbf{T}^{P_j} \qquad (r = 1, \ldots, n) \tag{A.39}$$

to yield:

$$F_1 = -K_1 z - b\dot{z} - \frac{\partial}{\partial x}\left[EI\frac{\partial^2 y_x}{\partial x^2}\right]\bigg|_{x=0} - \frac{\partial^2}{\partial x^2}\left[EI\frac{\partial^2 y_x}{\partial x^2}\right] \tag{A.40}$$

$$F_2 = -\frac{\partial^2}{\partial x^2}\left[EI\frac{\partial^2 y_x}{\partial x^2}\right]\Delta x \tag{A.41}$$

$$F_3 = \frac{\partial}{\partial x}\left[EI\frac{\partial^2 y_x}{\partial x^2}\right]\bigg|_{x=L} \tag{A.42}$$

$$F_4 = EI\frac{\partial^2 y_x}{\partial x^2} \tag{A.43}$$

Step 4.i The equations of motion are assembled according to

$$F_r + F_r^* = 0 \qquad (r = 1, \ldots, n) \tag{A.44}$$

to yield the same four equations (A.21) - (A.24) as those from the Lagrangian section.

A.6 Completing the Equations of Motion

Once the equations for the analogous rigid particle and body system are derived using one of the above techniques (or the same extension applied to some other technique), the equations must be altered to be correct for the continuous system. These n equations of motion include the natural boundary conditions for the problem. As stated above, each term that contains a contribution from an elemental particle can be altered into an equivalent term for the continuous system. Since each particle contributes to the dynamical equations in the same way as all the other particles in the continuous member, superposition can be used to alter the equations.

Step 5 *The equations which correspond to continuous coordinates are considered first.* For the example in the paper, equation (A.45) is the only equation of this type. This

equation was derived in the Lagrangian section as equation (A.22) (following from (A.18) for $\xi_i = y_x$). The same equation (A.45) was derived in the Kane section from (A.36) and (A.41) which correspond to the partial velocities in the y_x direction.

The equations will have terms with differentials (for example, Δm_x and Δx). The differentials should be divided through and the limit should be taken as the differentials approach zero. The motivation for this is that y_x represents an infinity of coordinates distributed throughout the continuum. The equations derived for these coordinates apply at each location in the continuum. The fact that there exists many of these particles does not alter the local scope of these equations. Example equation (A.22) is divided by Δx and in the limit becomes:

$$\rho_x \left(\frac{\partial^2 y_x}{\partial t^2} + \frac{d^2 z}{dt^2} \right) + \frac{\partial^2}{\partial x^2} \left[EI \frac{\partial^2 y_x}{\partial x^2} \right] = q(x) \tag{A.45}$$

Step 6 *In each of the remaining equations, terms that have no reference to the beam particles are left unchanged. Terms that contain any of the physical properties of the continuous particle must be altered.* Using the principle of superposition, this particle represents a series of particles. To correct this term, it should be summed (in practice, integrated) over the continuum. Differential quantities are converted to densities so that this integration can be performed.

For the example, only one term in the remaining equations contains a reference to the particle's mass, Δm_x:

$$\Delta m_x \left(\frac{\partial^2 y_x}{\partial t^2} + \frac{d^2 z}{dt^2} \right)$$

This differential is converted to ρ_x and the term must be integrated because it refers to a series of particles, not to just a single particle. The result is:

$$\int_0^L \rho_x \left(\frac{\partial^2 y_x}{\partial t^2} + \frac{d^2 z}{dt^2} \right) dx$$

The final set of partial differential equations is:

$$\int_0^L \rho_x \left(\frac{\partial^2 y_x}{\partial t^2} dx + \frac{d^2 z}{dt^2} \right) + M_a \frac{d^2 z}{dt^2} + K_1 z = -\frac{\partial}{\partial x} \left[EI \frac{\partial^2 y_x}{\partial x^2} \right] \Bigg|_{x=0} \tag{A.46}$$

$$\rho_x \left(\frac{\partial^2 y_x}{\partial t^2} + \frac{d^2 z}{dt^2} \right) + \frac{\partial^2}{\partial x^2} \left[EI \frac{\partial^2 y_x}{\partial x^2} \right] = q(x) \tag{A.47}$$

$$K_2 w = \frac{\partial}{\partial x} \left[EI \frac{\partial^2 y_x}{\partial x^2} \right] \Bigg|_{x=L} \tag{A.48}$$

$$\frac{\partial^2 y_x}{\partial x^2} \Bigg|_{x=L} = 0 \tag{A.49}$$

A.7 Conclusions

A simplified method for obtaining the exact partial differential equations of motion for systems that contain continuous elements has been presented. This method can be utilized in conjunction with any standard method for deriving equations of motion of systems. The technique is particularly useful in Kane's method because it enables Kane's method to be conveniently used on a class of problems that it previously could not solve without the use of an assumed-mode shape approximation. The most significant benefit of the technique is that it requires much less mathematical manipulation than the standard approaches.

A.8 Implementation of Kane's Method Version in MACSYMA

Kane's Method and the new extension to Kane's method were implemented in MACSYMA in order to automate the equation generating process. The MACSYMA source code is listed in Appendix B. The built-in MACSYMA functions make the manipulation of the mathematics simple. Two of the three pages of code which implement the method

essentially define some vector operations. The last page of code provides functions which perform each of the basic steps of Kane's method.

The user performs the following steps:

- Define the number of reference frames.
- Define the relative rotations of the frames.
- Define the rotation rates of the various frames.

The user needs only supply a definition of each frame with respect to any other frame. As long as the information provided is sufficient to determine the missing transformations. For example, if there are three frames, a, b, and c, the user can define the transformation of b in a, and the transformation of c in b. The code will automatically generate the transformation of c in a when the function `fill_in_trans_matrix()` is called.

The process continues exactly as it would be performed on paper:

- The number of linear and rotary speeds are defined.
- The speeds are defined in terms of the coordinates chosen.
- The number of masses and inertias are specified.
- The masses and inertias are defined in terms of the coordinates chosen.
- The velocity of each mass is specified as a vector quantity using the chosen coordinates.
- The number of forces and torques is specified.
- The force and torque vectors are specified in terms of the coordinates chosen.
- The velocities of the forces and torques are specified.

- Now the functions of Kane's method are called in order:
  - **`compute_general_forces();`** — computes the generalized forces.
  - **`compute_inertia_forces();`** — computes the inertia forces.
  - **`compute_fstar();`** — computes the generalized inertia forces.
  - **`compute_equations();`** — assembles the equations of motion into variable "eqn".

The partial velocities never need to be calculated because that step is automatically performed by the computer. Fortran code (or lisp-like code) output of the equations is then available as a standard feature of MACSYMA.

MACSYMA Code

Appendix B

This Appendix gives the MACSYMA code that was used to generate the equations of motion with Kane's Method. The first file contains the basic function definitions.

```
/* -*- Mode: Macsyma; -*- */

remarray(genf,vmpart,vfpart,amass,fstar,eqn);
matchdeclare([coef,dir1a,dir1b,dir1c,frame1,
                   dir2a,dir2b,dir2c,frame2],true);

defmatch(dyadtest,coef*dy(uvect(dir1a,dir1b,dir1c,frame1),
      uvect(dir2a,dir2b,dir2c,frame2)));
defmatch(vecd,coef*uvect(dir1a,dir1b,dir1c,frame1));


uvect(aa,ss,dd,ff)::=buildq([aa,ss,dd,ff,
                             tem:sqrt(aa^2+ss^2+dd^2)],
      if tem = 0 then 0
      else buildq([a1:aa/tem,s1:ss/tem,d1:dd/tem,f1:ff],
                  uvect(a1,s1,d1,f1)));

vdot(a,b):=a[1,1]*b[1,1]+a[1,2]*b[1,2]+a[1,3]*b[1,3];
vcross(a,b):=uvect(a[1,2]*b[1,3]-a[1,3]*b[1,2],
                   a[1,3]*b[1,1]-a[1,1]*b[1,3],
                   a[1,1]*b[1,2]-a[1,2]*b[1,1],1);

/* This is the top level vector crossing routine
```

```
    each expression must be in the form of uvect's  */
cross(ex1,ex2):=block([exp1,exp2],
    exp1:expand(ex1),
    exp2:expand(ex2),
    if exp1=0 or exp2 =0 then 0 else
        if part(exp1,0)="+" then
            map(lambda([u],
            if part(exp2,0)="+" then
                map(lambda([v],ncross(v,u)),exp2)
            else ncross(u,exp2)),exp1)
        else if part(exp2,0)="+" then
                map(lambda([v],ncross(exp1,v)),exp2)
             else ncross(exp1,exp2));

/*  This pattern matches the vectors, transforms vectors
        to the same coordinate frame, and then calls the
        low level vcross function */

ncross(exp1,exp2):=block([p1,p2,coef1],
      (if vecd(exp1)#false then
        (p1:[dir1a,dir1b,dir1c].trans[1,frame1],
         coef1:coef)
       else if dyadtest(exp1)#false then
        (p1:[dir2a,dir2b,dir2c].trans[1,frame2],
          coef1:coef*uvect(dir1a,dir1b,dir1c,frame1))
        else error("wrong type of arg to ncross",exp1),
      if vecd(exp2)#false then
            p2:[dir1a,dir1b,dir1c].trans[1,frame1]
      else if dyadtest(exp2)#false then
            (p2:[dir1a,dir1b,dir1c].trans[1,frame1],
         coef:coef*uvect(dir2a,dir2b,dir2c,frame2))
        else error("wrong type of arg to ncross",exp2),
      coef*coef1*vcross(p1,p2)));

/* This is the top level vector dotting routine
    each expression must be in the form of uvect's  */
dot(ex1,ex2):=block([exp1,exp2],
    exp1:expand(ex1),
    exp2:expand(ex2),
    if exp1=0 or exp2 =0 then 0 else
        if part(exp1,0)="+" then
            map(lambda([u],
```

```
            if part(exp2,0)="+" then
               map(lambda([v],ndot(v,u)),exp2)
            else ndot(u,exp2)),exp1)
        else if part(exp2,0)="+" then
                 map(lambda([v],ndot(exp1,v)),exp2)
             else ndot(exp1,exp2));

ndot(exp1,exp2):=block([p1,p2,coef1],
      (if vecd(exp1)#false then
        (p1:[dir1a,dir1b,dir1c].trans[1,frame1],
         coef1:coef)
       else if dyadtest(exp1)#false then
        (p1:[dir2a,dir2b,dir2c].trans[1,frame2],
         coef1:coef*uvect(dir1a,dir1b,dir1c,frame1))
         else error("wrong type of arg to ndot",exp1),
       if vecd(exp2)#false then
             p2:[dir1a,dir1b,dir1c].trans[1,frame1]
       else if dyadtest(exp2)#false then
             (p2:[dir1a,dir1b,dir1c].trans[1,frame1],
         coef:coef*uvect(dir2a,dir2b,dir2c,frame2))
        else error("wrong type of arg to ndot",exp2),
       coef*coef1*vdot(p1,p2)));


vdiff(ex):=block([exp],
  exp:expand(ex),
  if part(exp,0)="+" then
        vdiff(first(exp))+vdiff(rest(exp))
  else   if vecd(exp)#false then
    if rot[frame1]=0 or subvarp(rot[frame1]) then
         diff(coef,t)*uvect(dir1a,dir1b,dir1c,frame1)
      else diff(coef,t)*uvect(dir1a,dir1b,dir1c,frame1)+
      cross(rot[frame1],exp)
     else error("wrong type of arg to vdiff",exp));

vers(th):=1-cos(th);

rot(k,th):=matrix([k[1]^2*vers(th)+cos(th),
      k[2]*k[1]*vers(th)-k[3]*sin(th),
      k[3]*k[1]*vers(th)+k[2]*sin(th)],
       [k[1]*k[2]*vers(th)+k[3]*sin(th),
      k[2]^2*vers(th)+cos(th),
```

```
      k[3]*k[2]*vers(th)-k[1]*sin(th)],
       [k[1]*k[3]*vers(th)-k[2]*sin(th),
      k[2]*k[3]*vers(th)+k[1]*sin(th),
      k[3]^2*vers(th)+cos(th)]);

init_trans_matrix():=(trans:zeromatrix(frames,frames),
                         trans[1,1]:ident(3));


find(x,n):=block([m],
  if x=[] then error("axis undefined",n) else
    if first(rest(first(x)))=n then
    (m:first(first(x)),trans[1,n]:scanmap(trigreduce,
        trans[1,m].trans[m,n]))
    else find(rest(x),n));

fill_in_trans_matrix():=(trans_list:[],
    for i:1 thru frames do
    for j:2 thru frames do
      if trans[i,j]#0 then
            trans_list:cons([i,j],trans_list),
    for i:2 thru length(trans_list)+1 do
            find(trans_list,i));

gentrans(f1,f2,vector,theta):=
            trans[f1,f2]:rot(vector,-theta);

vmpart[type,num1,dir,num2]:=
           diff(expand(vel_mass[type,num1]),u[dir,num2]);

vfpart[type,num1,dir,num2]:=
           diff(expand(vel_force[type,num1]),u[dir,num2]);

rectf(count,icnt):=(if icnt<= lin_velocities then
           (dir:1,
            n:icnt)
        else
           (dir:2,
            n:icnt-lin_velocities),
      if count=0 then 0 else
          (if count <= lin_forces then
       (type:1,
```

```
        num:count)
           else
        (type:2,
        num:count-lin_forces),
        dot(vfpart[type,num,dir,n],force[type,num])
          +rectf(count-1,icnt)));

rectfstar(count,icnt):=(if icnt<= lin_velocities then
            (dir:1,
             n:icnt)
         else
            (dir:2,
             n:icnt-lin_velocities),
  if count=0 then 0 else
            (if count <= lin_masses then
           (type:1,
    num:count)
      else
                 (type:2,
    num:count-lin_masses),
   dot(vmpart[type,num,dir,n],
                                  inertia_force[type,num])
            +rectfstar(count-1,icnt)));

compute_general_forces():= (genf:[],
      for i:1 thru lin_velocities+rot_velocities
                 do
        genf:endcons(rectf(lin_forces+rot_forces,i),genf));

compute_inertia_forces():=for i:1 thru lin_masses+rot_masses
                  do (if i<= lin_masses then
           (type:1,
            inertia_force[type,i]:
                        -m[type,i]*vdiff(vel_mass[type,i]))
        else if i<= lin_masses+rot_masses then
                (num:i-lin_masses,
           type:2,
           ww:vdiff(vel_mass[type,num]),
           inertia_force[type,num]:
                       dot(-ww,
                      m[type,num])-
                (-cross(ww,dot(m[type,num],ww)))));
```

```
compute_fstar():= (fstar:[],
      for icount:1 thru lin_velocities+rot_velocities
                 do
        fstar:endcons(
            rectfstar(lin_masses+rot_masses,icount),fstar));


compute_equations():=(eqn:[],
    for icount:1 thru lin_velocities+rot_velocities
                 do
       (temp:ev(scanmap(trigreduce,
                   subst(sum,sums,fstar[icount])+
             subst(sum,sums,genf[icount])),eval),
            eqn:endcons(ev(temp,diff),eqn)));


kane:true;
```

This File uses the functions defined above to generate the equations of motion for the space craft shown in figure B.1 at the end of this appendix.

```
/* -*- Mode: Macsyma -*- */

remarray(rot,u,m,vel_mass,vel_force,force);

if kane#true then load("kane-functions.source");

/*   example begins here

        */
depends([x1,x2,y,th],t);
depends(y_dir,x_dir);

frames:2;      /* one is newtonian, one is rotating */

n:1;        /* newtonian */
a:2;        /* rotating frame a */

init_trans_matrix();

gentrans(n,a,[0,0,1],0);  /* initial rotation
                             of a in n */
        /*    0 radians around vector [0,0,1] in n */

/*  rot[n]  specifies the rotation rate
        of frame n as a vector */

rot[a]:diff(th,t)*uvect(0,0,1,n); /* rotation rate of
                                     a frame in n */
        /* dth/dt in the [0,0,1] direction */

fill_in_trans_matrix();

/*    u[1,n] is the nth linear velocity
      u[2,n] is the nth rotary velocity
*/

/* velocities */
```

```
lin_velocities:3;    /* dx1/dt dx2/dt dy/dt*/
rot_velocities:1;    /* just dth/dt */

        /* u[x,y] means y'th velocity of the x type */
        /* x = 1 linear */
        /* x = 2 rotary */
u[1,1]:diff(x1,t);
u[1,2]:diff(x2,t);
u[1,3]:diff(y,t);
u[2,1]:diff(th,t);


/* masses  */

lin_masses:2;         /* Big Mass plus little mass*/
rot_masses:1;         /* inertia of big mass*/


        /* m[x,y] means y'th mass of the x type */
        /* x = 1 linear */
        /* x = 2 rotary */
m[1,1]:ma;
m[1,2]:mb;
m[2,1]:j33*dy(uvect(0,0,1,1),uvect(0,0,1,1));

        /* vel_mass[x,y] means the velocity */
        /*  of mass m[x,y] in newtonian */
        /*   reference frame */
        /*   uvect(n1,n2,n3,frame) means */
        /*    a vector in the [n1,n2,n3] direction */
        /*    in reference frame frame */
vel_mass[1,1]:u[1,1]*uvect(1,0,0,a)+u[1,2]*uvect(0,1,0,a);
vel_mass[2,1]:u[2,1]*uvect(0,0,1,a);
vel_mass[1,2]:(u[1,1]-u[2,1]*y)*uvect(1,0,0,a)+
    (u[1,2]+(b+l)*u[2,1]+u[1,3])*uvect(0,1,0,a);


/* forces */
          /* */
lin_forces:2;
rot_forces:1;
```

```
        /* vel_force[x,y] means the velocity of */
        /*  of force acting on [x,y] in newtonian */
        /*   reference frame */
        /* x = 1 linear */
        /* x = 2 rotary */

          /* lin vel at Q */
vel_force[1,1]:u[1,1]*uvect(1,0,0,a)+(u[1,2] +
    b*u[2,1])*uvect(0,1,0,a);
          /* rot vel at Q */
vel_force[2,1]:u[2,1]*uvect(0,0,1,a);
          /* lin vel at P */
vel_force[1,2]:(u[1,1]-u[2,1]*y)*uvect(1,0,0,a)+
    (u[1,2]+(b+1)*u[2,1]+u[1,3])*uvect(0,1,0,a);

        /* force[x,y] is the force acting */
        /*  on  m[x,y] in newtonian */
        /*   reference frame */

          /* lin force at Q*/
force[1,1]:-ei*pdiff(y_dir,x_dir,3,0)*uvect(0,1,0,a);
          /* rot torque at Q */
force[2,1]:ei*pdiff(y_dir,x_dir,2,0)*uvect(0,0,1,a);
          /* force at P */
force[1,2]:-pdiff(ei*pdiff(y_dir,x_dir,2),x_dir,2)*
              uvect(0,1,0,a);


compute_general_forces();
compute_inertia_forces();
compute_fstar();
compute_equations();
}
```

B.1 Lagrange's Equations in MACSYMA

For reference, The following MACSYMA script for implementing Hamilton's Principle is given. The user inputs a T^* and a V. The equations of motion are returned.

```
/* -*- Mode: Macsyma -*- */
/* THIS FILE COMPUTES THE EQUATIONS OF MOTION OF A SYSTEM
    TS IS THE EXPRESSION FOR THE KINETIC COENERGY
    V IS THE EXPRESSION FOR THE POTENTIAL ENERGY
    X IS A LIST OF THE VARIABLES  IN THE FORM [X1,X2,X3...]
    YOU MUST SET DEPENDANCIES OF YOUR VARIABLES WITH TIME
    USAGE:    DEPENDS(X,T);
    TS:1/2*M*DIFF(X,T);
    V: 1/2*K*X^2;
    LAGRANGE(TS,V,[X]);

  NOTE: X IS DECLARED DEPENDENT ON T BUT IS USED AS X NOT
      X(T) !!    */

lagrange(ts,v,x):= (depends(x,[t]),
   if x#[] then expand(append([lag(ts,v,part(x,1))],
  lagrange(ts,v,rest(x)))
  else []);
lag(ts,v,x):=diff(diff(ts-v,'diff(x,t)),t)-diff(ts-v,x);
```

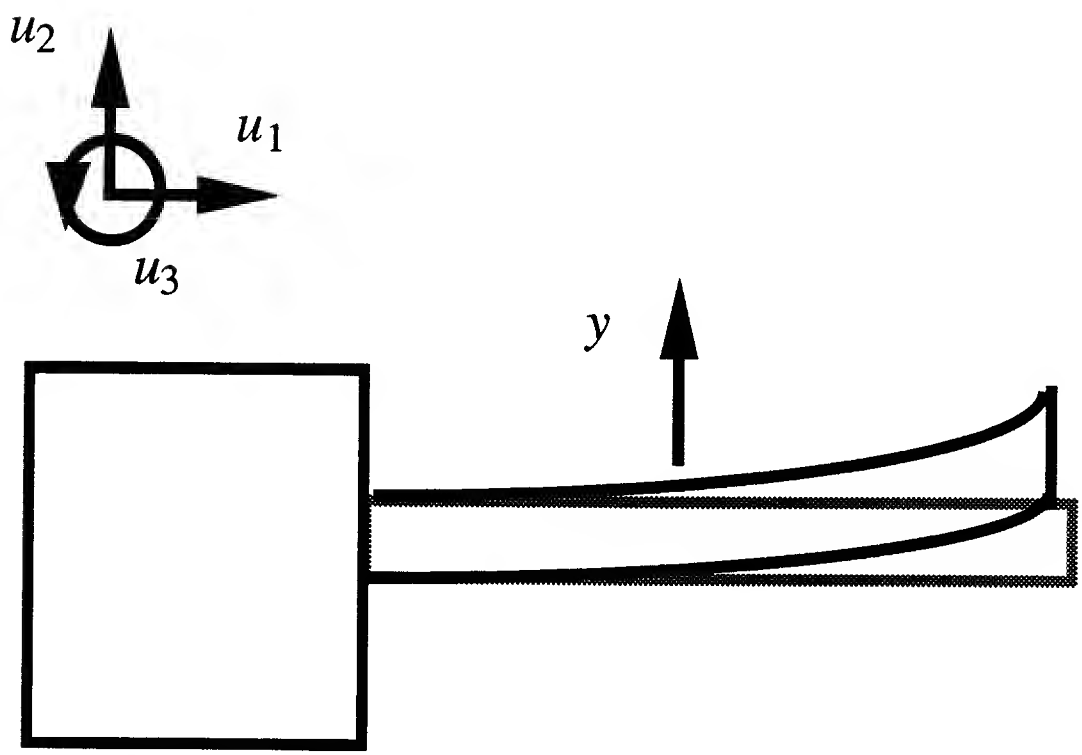


Figure B.1: Spacecraft Example from Thomas Kane's book, **Spacecraft Dynamics** [65]

Characterization of the RMS Workspace

Appendix C

In order to use the DRS shuttle simulator described in section 2.3 as a test system, its behavior must be known. First, the system was analyzed in terms of oscillation frequency. The vibrational data used in this paper were generated by placing the arm in a position in the workspace, exciting the arm and monitoring the motion of the endpoint. The excitation was an initial velocity in several of its base joints at $t = 0$ with the axes at their set points. This corresponds to an impulse in current (torque) in these joints.

The endpoint vibration in the three principle directions (x,y,and z in the orbiter body coordinates) was recorded. Each of these three signals was processed by subtracting the mean value so that there was no DC component to the signal. The data were then windowed using a cosine taper function [101]. The discrete Fourier transform of this processed signal yielded the fundamental frequency components of the arm endpoint vibration in the three orbiter coordinate directions (Figure 2.3). Only the first two modes were examined for each configuration of the arm and conclusions generated. Figures C.1 and C.2 show two examples of this data.

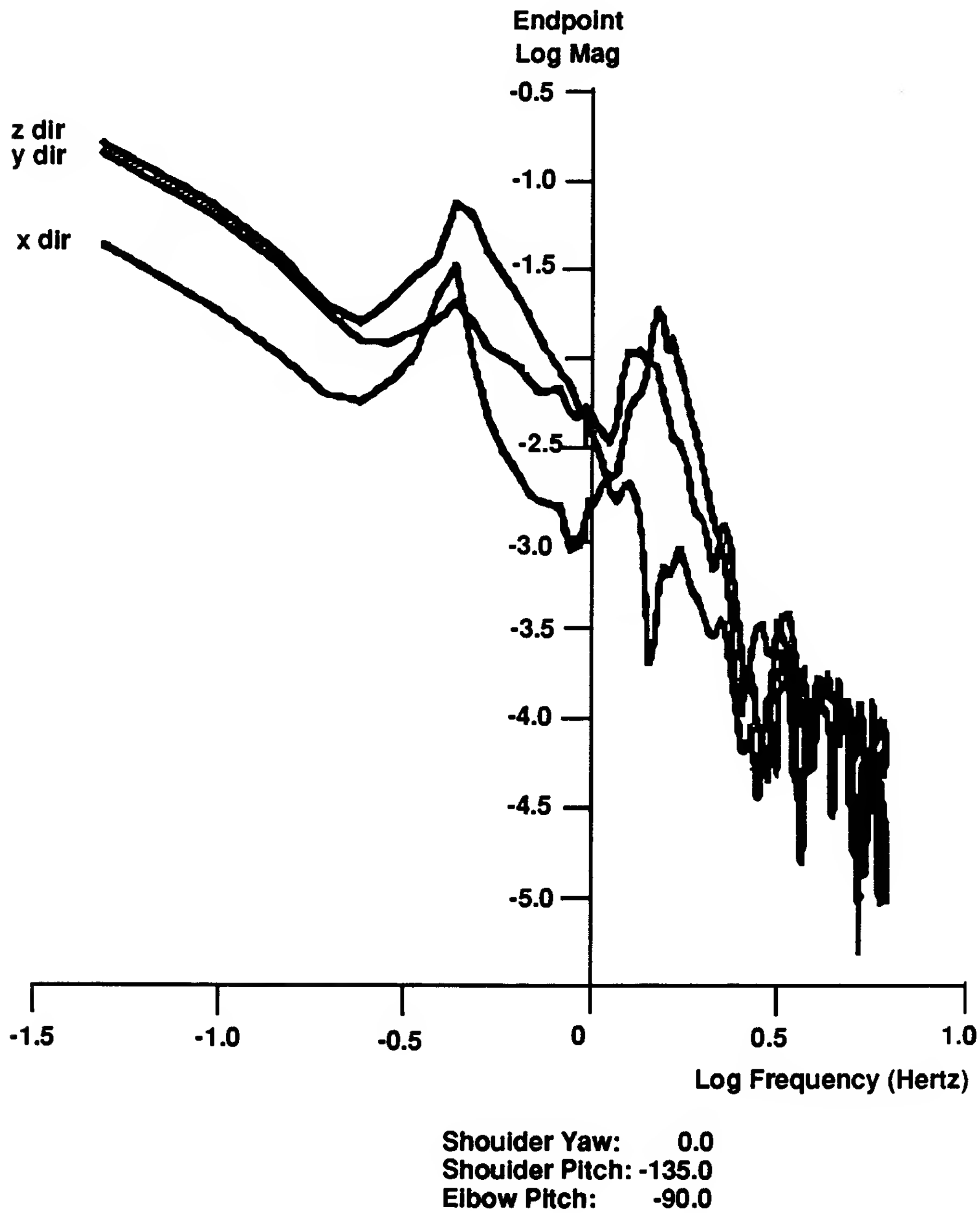
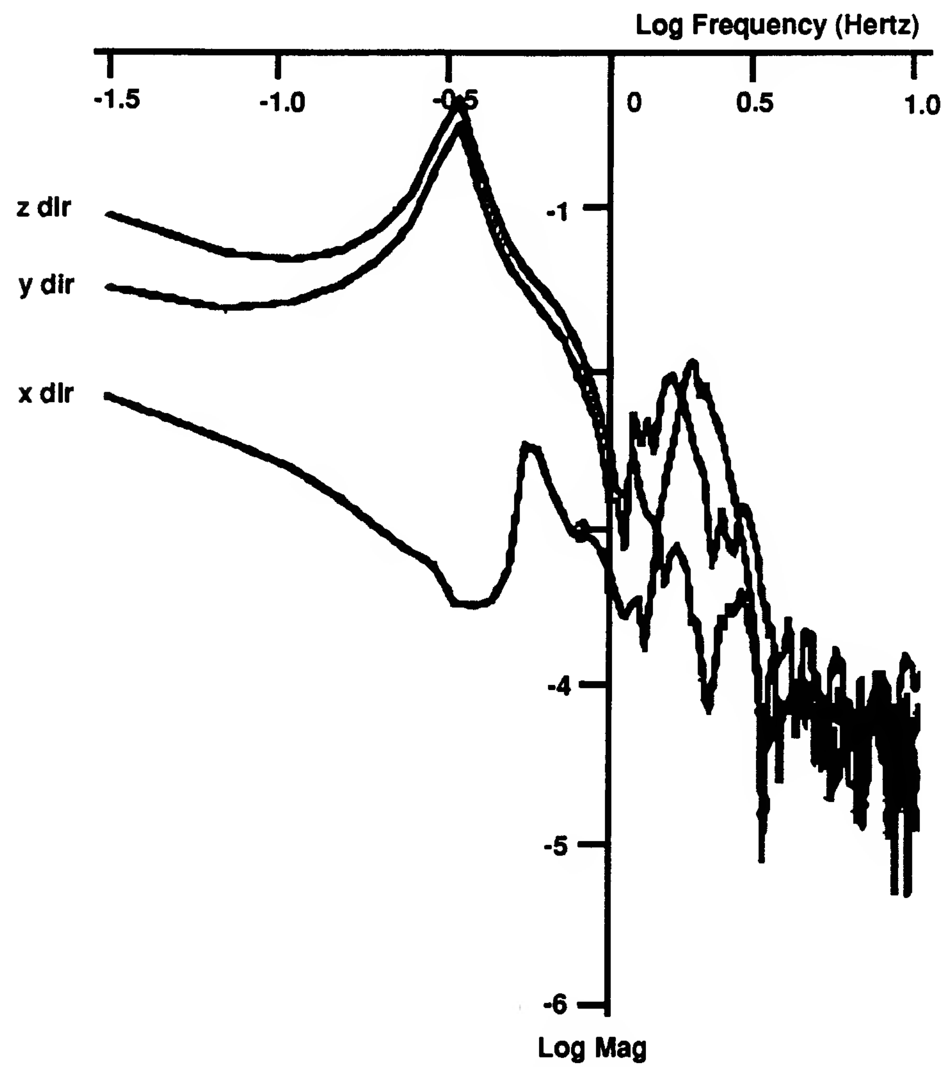


Figure C.1: Frequency content of endpoint vibration for the RMS positioned along the longeron (shoulder yaw at 0°) with a shoulder pitch of 135° and an elbow pitch of −90°.



Shoulder Yaw: 90.0
Shoulder Pitch: 135.0
Elbow Pitch: -90.0

Figure C.2: Frequency content of endpoint vibration for the RMS positioned perpendicular to the longeron (shoulder yaw at 90°) with a shoulder pitch of 135° and an elbow pitch of −90°.

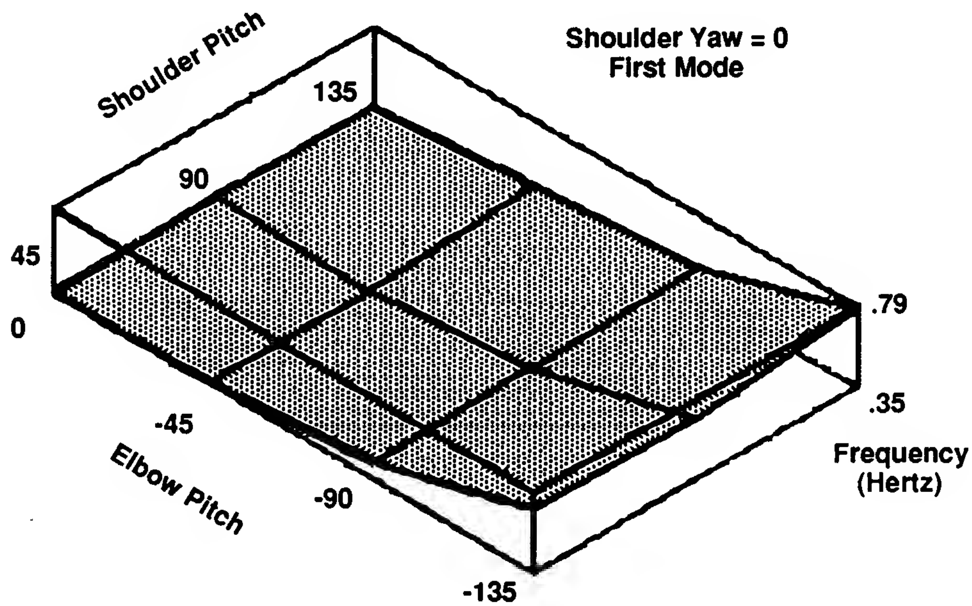


Figure C.3: The first mode of the RMS as a function of shoulder pitch and elbow pitch. Shoulder yaw is fixed at 0° (the arm is moving in a vertical plane which includes the longeron).

Figures C.3, C.4, C.5, and C.6 show a summary of the vibrational characteristics of the RMS workspace. The lowest first natural frequency observed was .32 hertz when the arm was fully extended. The lowest second natural frequency was 1.1 hertz. The highest first mode observed was .8 hertz. The highest second mode observed was 3.2 hertz.

By examining the frequency response data, several patterns become apparent. As the shoulder yaw axis is moved (for fixed values of all other joint rotations) the frequencies vary over a ninety degree slice and are axi-symmetric. Therefore the arm has the same first two modes if it points either toward the tail of the orbiter or toward the nose. The same holds true for the arm pointing toward the port or the starboard.

The variation of this stiffness as the shoulder yaw axis moves through ninety degrees stems from the flexibility of the connection of the arm to the shuttle. The longeron to which the arm is attached is stiffer along its axis. When the arm is pointing along this

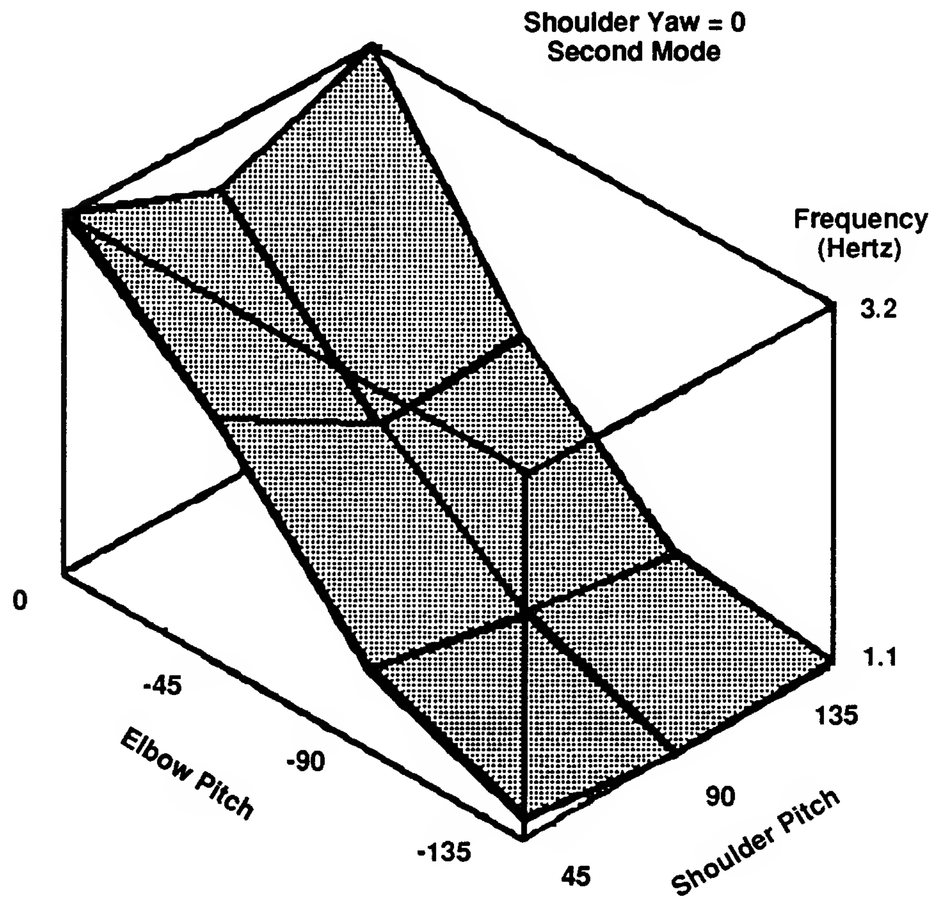


Figure C.4: The second mode of the RMS as a function of shoulder pitch and elbow pitch. Shoulder yaw is fixed at 0° (the arm is moving in a vertical plane which includes the longeron).

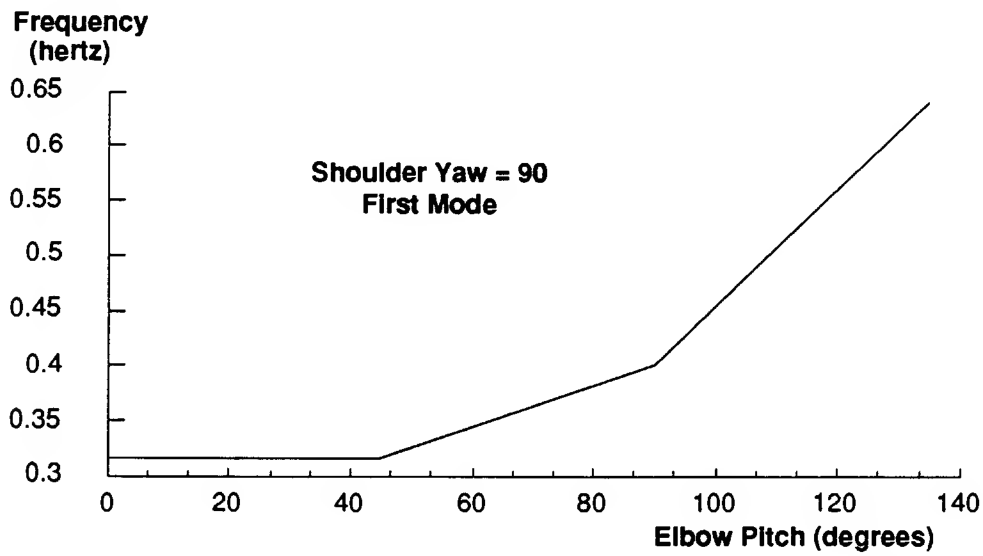


Figure C.5: The first mode of the RMS as a function of elbow pitch. The plot is shown in two dimensions because the frequencies are the same for all values of shoulder pitch. Shoulder yaw is fixed at 90° (the arm is moving in a vertical plane which is perpendicular to the longeron).

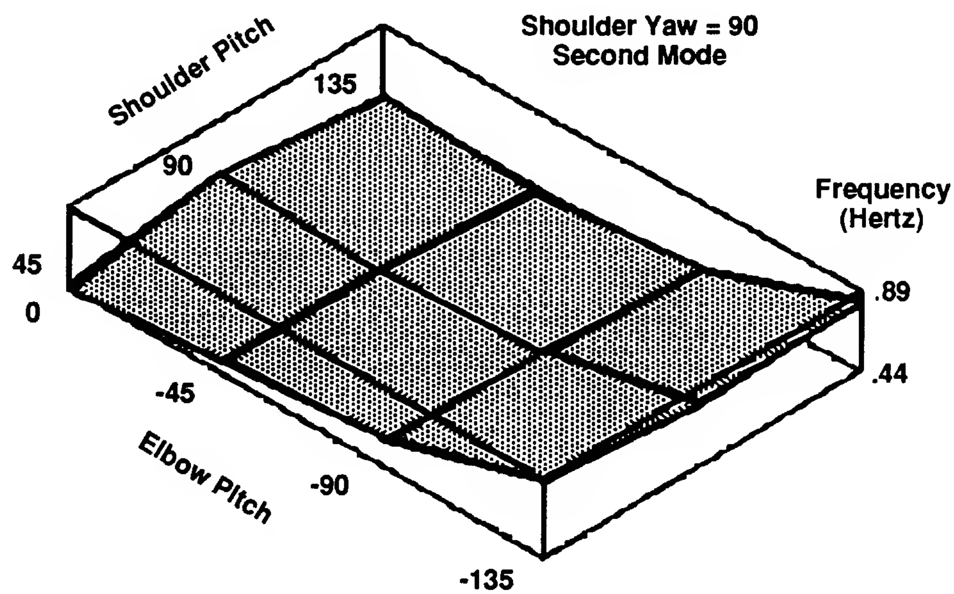


Figure C.6: The second mode of the RMS as a function of shoulder pitch and elbow pitch. Shoulder yaw is fixed at 90° (the arm is moving in a vertical plane which is perpendicular to the longeron).

axis (figure 1), the modes are higher than when it is off axis. When the arm shoulder yaw axis is set to zero degrees (pointing along the longeron), the vibration is at the same frequency in all three directions (figure C.1). As the shoulder yaw moves toward ninety degrees, the vibration of the RMS perpendicular to the plane of the arm stays the same while the vibration of the arm in the plane lowers in frequency (by as much as 40%). This can be seen in figure C.2.

Geometrically Varying Systems

Appendix D

The techniques presented in chapters 3 and 4 were derived for linear systems. Systems that significantly change vibration characteristics as they change in configuration are more difficult to command without exciting residual vibration. This appendix will address some of the issues of generating vibration-reducing commands for systems that have significant geometry changes. Because each nonlinear system poses its own, unique set of control difficulties, a truly general formulation is not possible with current tools. However, several specific nonlinear systems were considered and some conclusions can be made from these examples. This appendix presents some discussion on how command shaping applies to four different nonlinear situations. For some, heuristic methods for shaping commands were presented. The results are heuristic because no general nonlinear system theory exists.

Four types of systems will be considered:

- Systems for which the period of oscillation varies by a relatively small amount.
- Systems that experience large changes in period of oscillation but are velocity limited and therefore, most of their move is spent saturated at the maximum velocity.

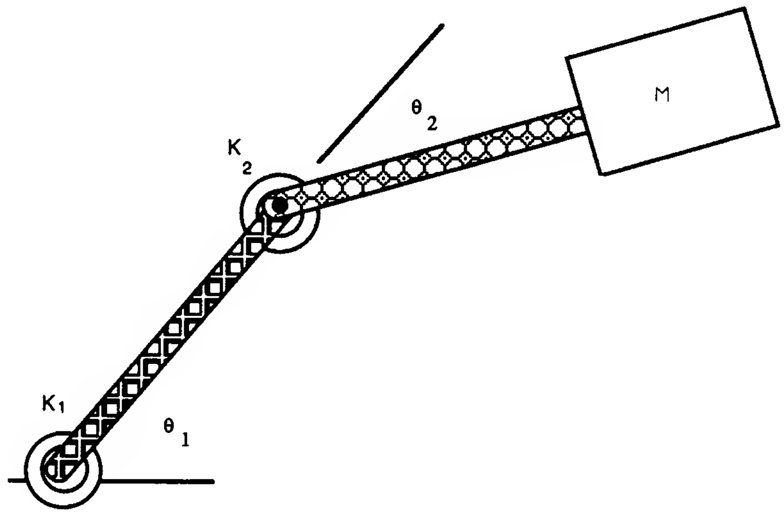


Figure D.1: Nonlinear test model used to evaluate various geometrically nonlinear techniques.

- Systems that suddenly change by a significant amount because of a discontinuity (for example, a manipulator picking up a payload).
- Systems that experience large changes in period of oscillation and spend a relatively small percentage of the move time at velocity saturation.

The techniques discussed throughout this chapter will be demonstrated on a two link model system shown in figure D.1. The system model is a full nonlinear simulation of a rigid link manipulator. The coriolis and centrifugal terms have been retained. The flexibility of the system is lumped in the joints. Chapter 4 showed some of these techniques used on the nonlinear shuttle simulator (DRS) with link flexibility.

D.1 Systems with Small Changes in Period

For systems that change oscillation period by a small amount (less than 20% or 30%) the robust sequences presented above may be acceptable. The robustness of the sequences accommodates frequency uncertainties or shifts of this amount. The two-link system shown in figure D.1 was used as a test system. The following parameters were used in the model:

- length of link 1 and 2: 1.0
- mass of link 1 and 2: 1.0
- inertia of link 1 and 2: 1.0
- end mass: 12.0
- end mass inertia: 9.0
- spring 1: 50.0
- spring 2: 100.0

Figure D.2 shows a move for which the system changes period of oscillation by approximately 24% (from .17 hz to .21 hz). The four impulse sequence used to shape the input signal was fixed for a system of .17 hz fixed frequency. The residual vibration for this system with no damping is 6.8%. Note that the inherent robustness enables the shaping to be effective for this significantly nonlinear system.

D.2 Systems that Velocity Saturate

A great many systems that significantly change their vibrational characteristics as they move are velocity limited. An example of this is the Space shuttle RMS. The 50-foot long RMS reaches velocity saturation after moving just a few inches. Inputs to these systems can be shaped just as in chapter 4 as long as the commanded trajectory (before shaping) accounts for the velocity saturation.

For example, the DRS uses velocity commands. The DRS system is accelerated up to maximum velocity with a vibration-reducing command. It slews at maximum velocity and is then decelerated with a vibration-reducing command. The impulse sequence used for accelerating the system is different from the sequence used to stop the system. Each is selected for the configuration of the system when it acts on the system. For example, if the space shuttle arm accelerates when its payload is close the shuttle it may have a

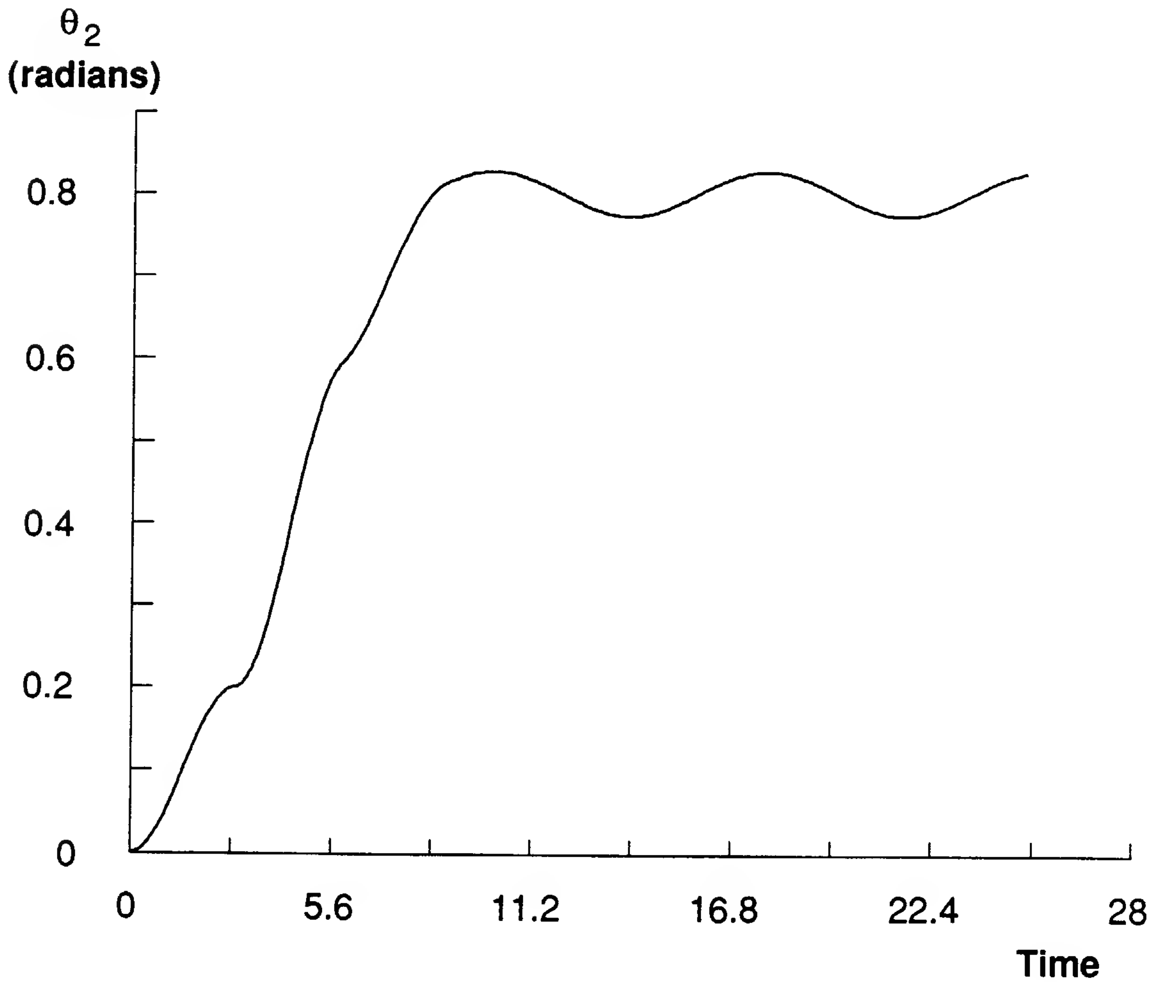


Figure D.2: Robust shaping with a nonlinear system. A four impulse sequence successfully drives the two link system shown in figure D.1. The parameters of the model were chosen so that the frequency shift is approximately 24%

frequency of approximately .5 hz. A sequence tuned for .5 hz is used to accelerate the arm. When the arm is commanded to stop by the operator, it may have a frequency of .2 hz. A sequence tuned for .2 hz is used to decelerate the arm.

Figure D.3 demonstrates the results of this approach on a highly nonlinear 2 link system (shown in figure 4.31). The system was altered to make it highly nonlinear. The following parameters were used:

- length of link 1: 1.0
- length of link 2: 3.0
- mass of link 1 and 2: 1.0
- inertia of link 1 and 2: 1.0
- end mass: 50.0
- end mass inertia: 40.0
- spring 1: 0.0
- spring 2: 30.0

These system parameters were chosen so that the two-link system changed from a linearized natural frequency of .48hz to .12hz during the 2 radian move (a factor of 4 change in frequency). The first (base) spring constant was set to zero so that only one, nonlinear mode was present. The one mode system demonstrates the vibration-reduction more clearly than a coupled, multiple-mode system.Damping was not included in the example in order to isolate the effect of the shaping. The residual vibration in this case was only 1.2% of the move distance.

Although these systems significantly change period of oscillation throughout a move, they do not change much during the acceleration or deceleration phase. For this reason shaping can readily be applied to these systems by generating a rough "map" of the frequency shifts. Each time the system is started or stopped, the "map" provides the estimated frequency which is used to generate the shaping sequence.

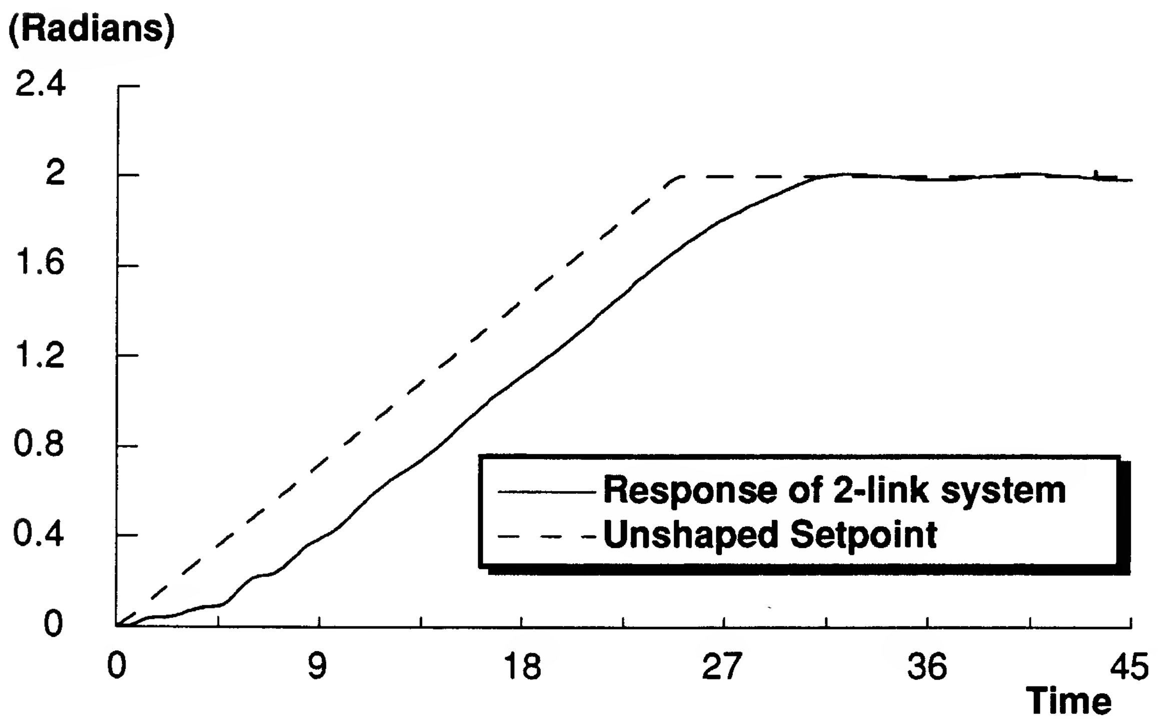


Figure D.3: Shaping inputs to a highly nonlinear system that is velocity limited. The two link system of figure D.1 was accelerated with one shaping sequence (.48hz) and decelerated with another (.12hz). The parameters of the model for this test were chosen so that the natural frequency of the system changed by a factor of four during the move.

D.3 Discontinuous Systems

Systems that undergo a sudden discontinuity must also be considered. An example of this is picking up a large payload. If the frequencies of the system are known both before and after the sudden change, the sequence used for shaping can be changed. This solution assumes a "quasi-static" approximation to the change can be used. For example, if the machine moves to a new location, stops, and grasps a heavy object; the frequencies of the system have suddenly changed. Fortunately, the shaping software can be made aware that the system has changed and can change the shaping sequences accordingly. If the machine is significantly accelerating or decelerating during the discontinuity (such as throwing a ball), some dynamic effects may interfere with the vibration reduction. Since the "dynamic" payload change situation is specialized and application specific, it is not discussed in greater detail.

D.4 Quasi-Linear Assumptions

The first requirement for deciding which approach to use for nonlinear systems is to examine the conditions under which the quasi-linear approximation (use of impulse sequences for shaping) holds. This section examines how well the three-impulse shaping sequence works on systems with different degrees of geometric nonlinearity. Figure D.1 shows the geometry of a test system that will be used for this analysis. Two different sets of parameters were chosen for this model in order to produce two unique systems from which the results in this section were generated. The full nonlinear model was simulated and prefiltered inputs were delivered to the two systems. Two types of input commands, steps, and versines were used to move the systems.

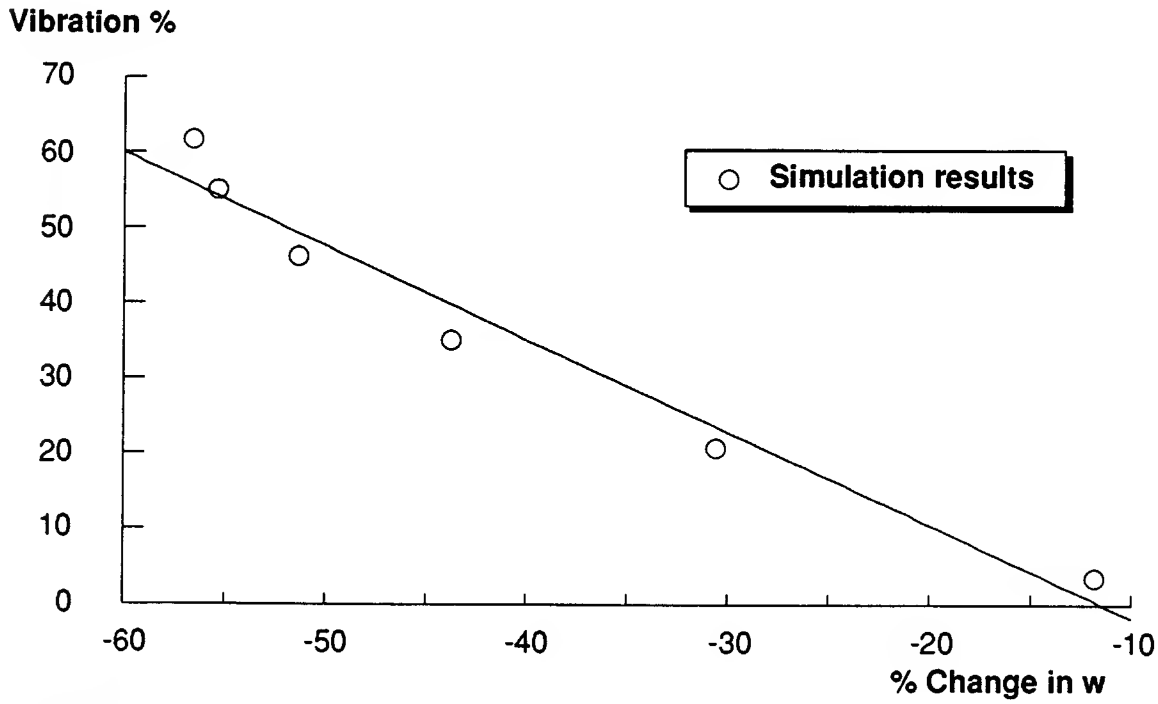
Figures D.4 D.5 and D.6 shows the amount of residual vibration that remains for various moves of the two different systems. The input is prefiltered with a sequence

designed for the linearized natural frequency of the system at the starting position of the move. The graphs of figures D.4, D.5, and D.5 indicate that linear regression seems to be reasonable method of estimating the residual vibration amplitude as a function of the frequency shift of the system. The slope of this line seems to depend on the parameters and type of system. This interesting behavior is one topic of suggested future research (see section 8.2).

D.5 Quasi-static System Tracking

Another approach to nonlinear systems is that of continuously tracking the oscillation period of the system with a shaping sequence. This approach involves developing a crude model of the system natural frequencies. (For example, a rough map of the frequencies as a function of position of the system). As the system moves, the impulse sequence can be changed based on the frequency of the system at the current position. This approach relies on an assumption that the shifts in frequency are "quasi-static". For many systems this assumption will hold.

Figure D.7 shows a schematic of this new approach (which will be refered to as the "tracking sequence" approach). Figure D.8 demonstrates the response of the nonlinear test system commanded using a tracking shaping sequence. The tradeoff with using this approach is that each time the shaping sequence is changed, a discontinuity is introduced in the output. This discontinuity will cause a vibration in the output. The discontinuities are small if the frequencies of the system have not changed significantly during the period of time that the filter was fixed. Figure 4.4 shows that the impulse sequences vary continuously as the frequency of the system changes. In the case of the continuously changing filter, the discontinuities are minimized. Several systems that significantly change oscillation period were generated by using different parameters in the model of figure D.1. All those that were tested using the tracking sequence approach



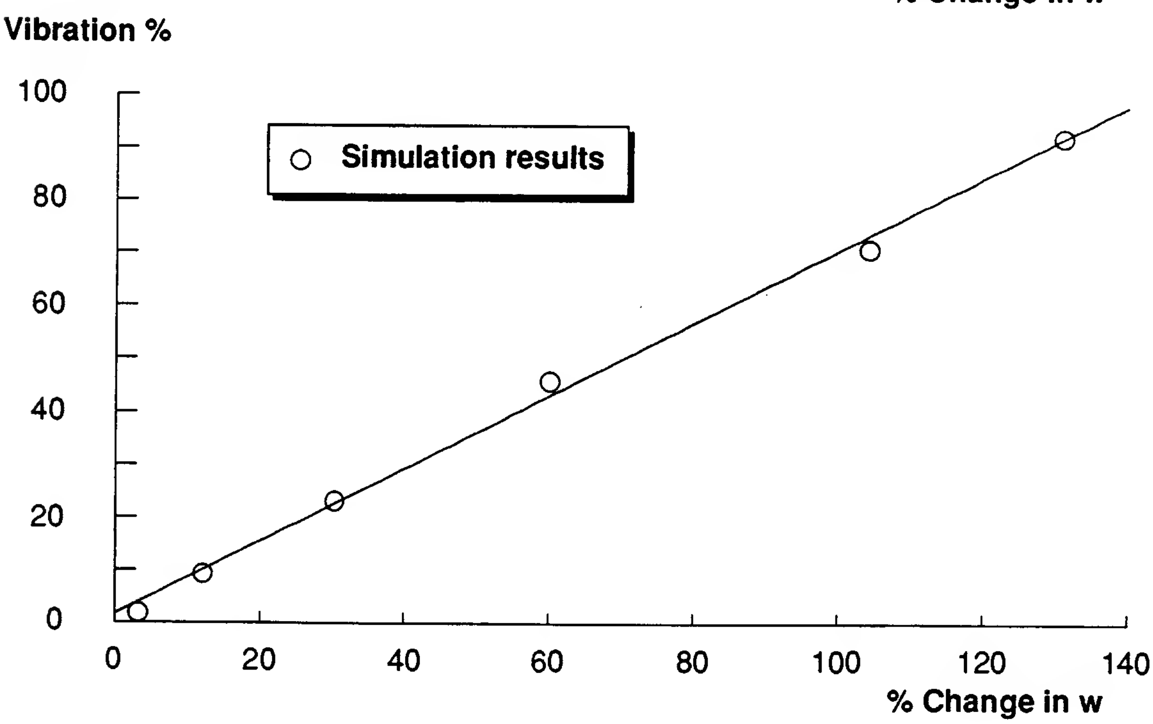


Figure D.4: Summary of residual vibration from nonlinear systems. Residual vibration percentage for the two link system. The top plot is a summary of residual vibration for a system commanded to step forward by different amounts. The bottom plot is the same system commanded to step backward.

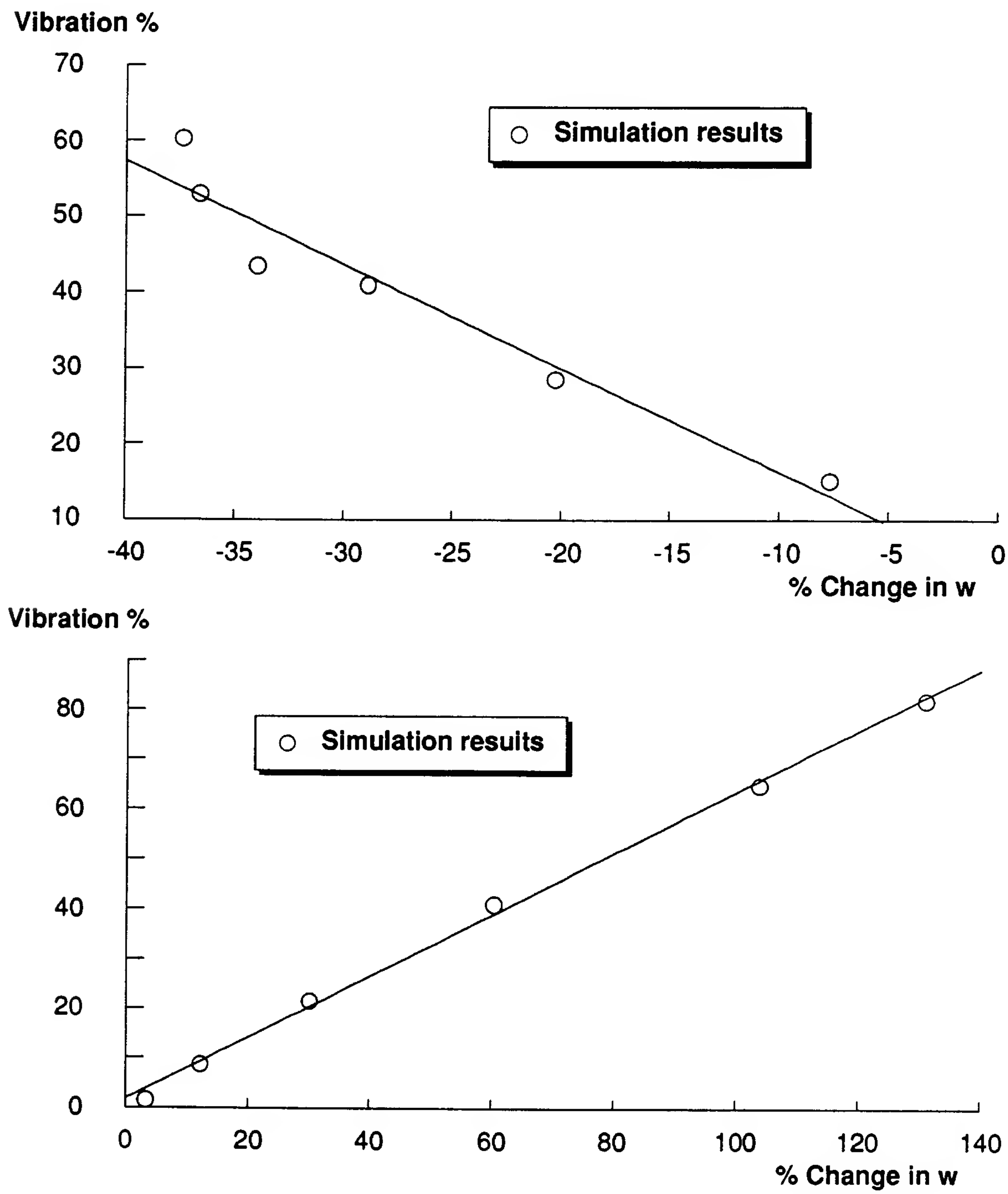


Figure D.5: Summary of residual vibration from nonlinear systems. Residual vibration percentage for the two link system. The top plot is a summary of residual vibration for a system commanded with different length versine trajectories in the forward direction. The bottom plot is the same system commanded with versine trajectories in the other direction.

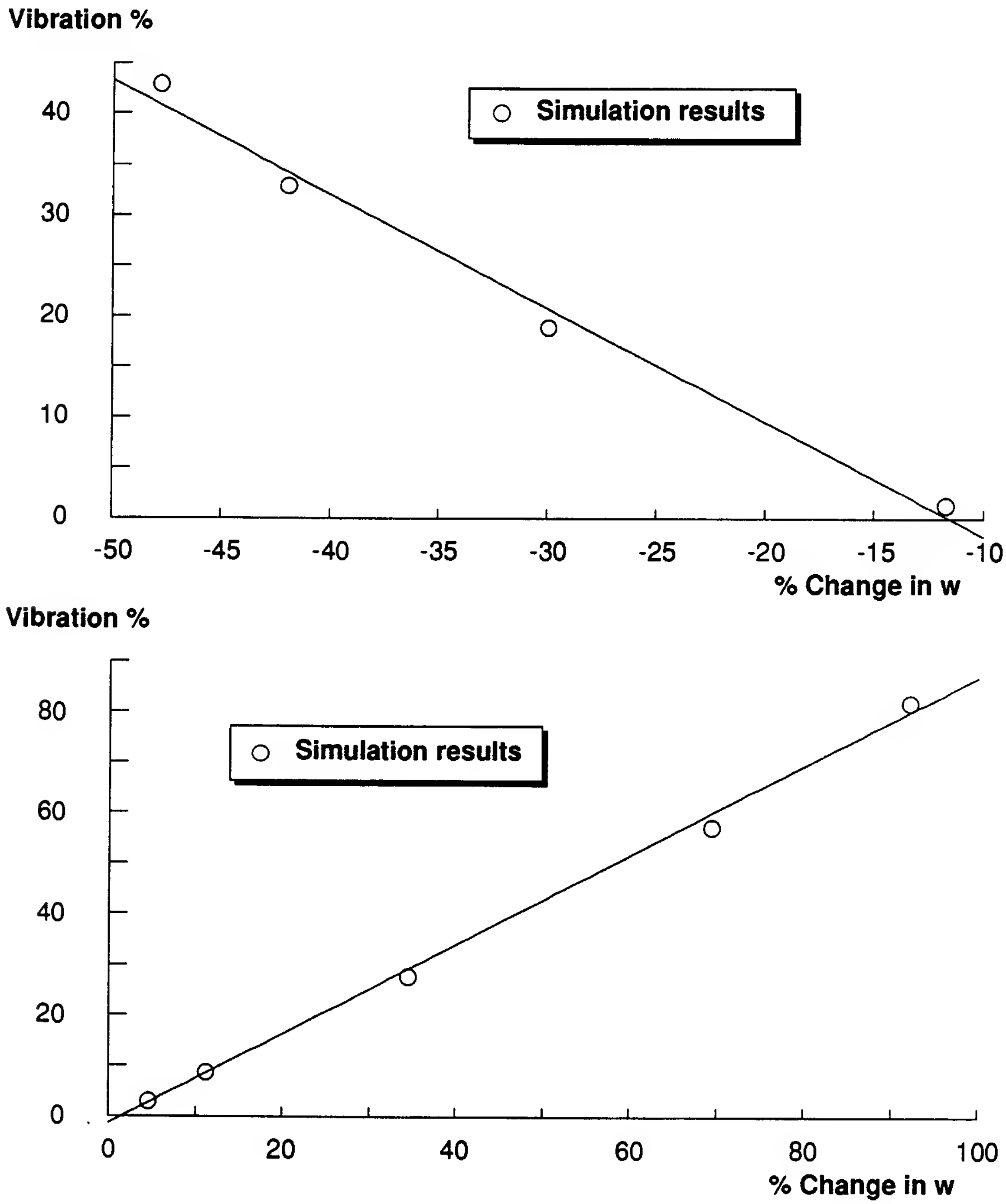


Figure D.6: Summary of residual vibration from nonlinear systems. Residual vibration percentage for a two link system. These results are for a system with drastically different parameters from that of previous two figures. The top plot is a summary of residual vibration for a system commanded to step forward by different amounts. The bottom plot is the same system commanded to step backward.

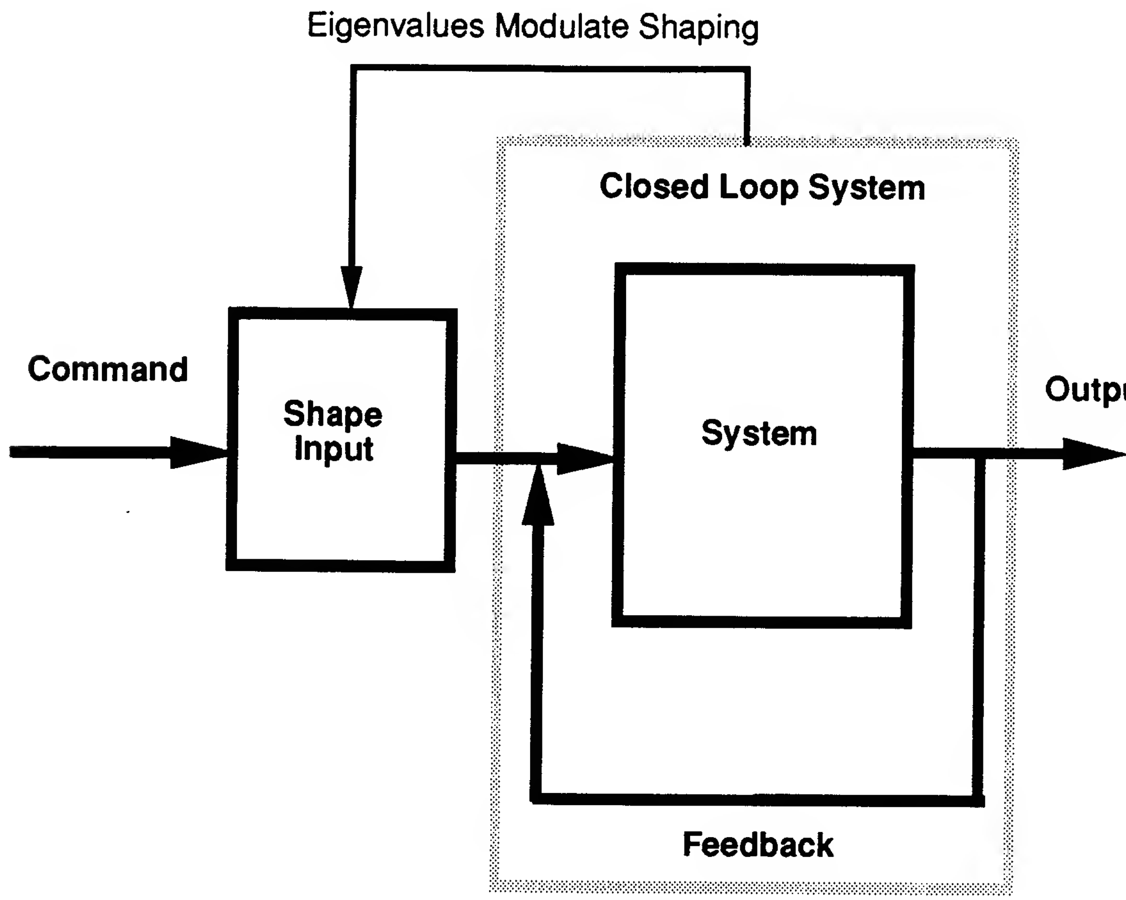


Figure D.7: Schematic diagram of the tracking sequence concept. The linearized eigenvalues were solved as a function of system position. These eigenvalue estimates were used as a rough frequency estimate for the command shaping.

were moved with reduced vibration. One example is given in figure D.8.

Table D.9 shows a summary of the effectiveness of tracking the frequencies of a two link, nonlinear simulation. Two types of inputs were used, a short versine input (20 second duration), and a long versine input command (40 second duration).

Five methods were attempted. The first three involved the use of one, stationary shaping sequence. Table D.9 shows that this approach is effective for reasonably small frequency shifts of the system. The second approach is to use the tracking-sequence approach presented in this section. The last approach was to discretize the space into 3 regions and suddenly change the filter as the arm moved from one region to another.

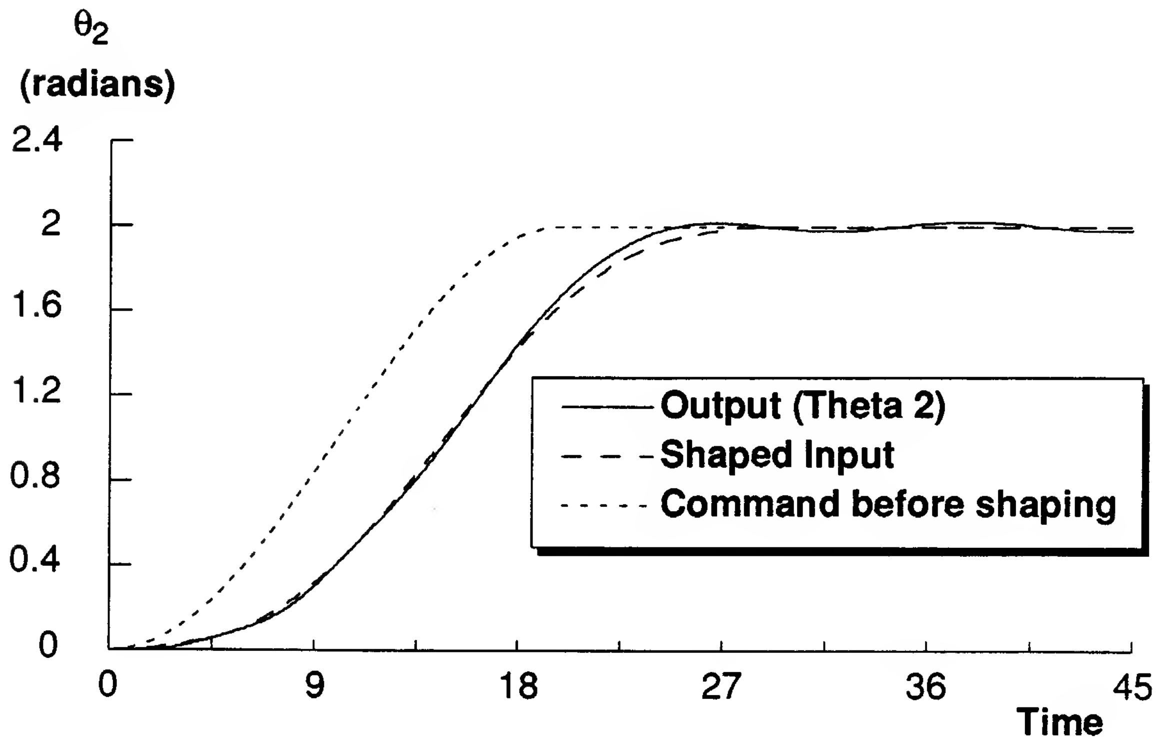


Figure D.8: Response of two-link system to a tracking shaping sequence. The sequence that shapes the input is updated during each servo cycle based on a linear estimate of the system eigenvalues.

Shaping Method	**Slow Versine (2 r in 40 Seconds)**	**Fast Versine (2 r 20 Seconds)**
One Sequence Set for Beginning	Amplitude: .5%	Amplitude: 5%
One Sequence Set for Midpoint	Amplitude: .5%	Amplitude: 5%
One Sequence Set for End	Amplitude: 2%	Amplitude: 7%
Sequence Continuously Tracking System	Amplitude: 2%	Amplitude: 7%
Three Separate Regions	Amplitude: 7%	Amplitude: 11%

Figure D.9: Summary of the effectiveness of nonlinear system frequency tracking. This is the summary of some sample results obtained from simulations of a 2-link system. Two inputs were sent to the system using each of the five strategies. The amplitude was the amplitude of residual vibration.

The 3 band approach did not work well because the sudden switching of the filters induced step discontinuities in the system input. These discontinuities generated residual vibration in the system.

The results of table D.9 are for a system without any damping, so the results can be used only for comparison. Real system damping will help eliminate small residual vibrations. In addition, vibrations induced at the end of the move will cause more problem in a real system than vibrations induced at the beginning of a move. Vibrational energy induced early in a move dissipates during the motion. These tests do not distinguish the point at which vibration is induced.

Bibliography

[1] **Abelow, Allen V.,**
"Dynamic Equation Set for a Simplified Simulation of the Space Shuttle Remote Manipulator System," *Charles Stark Draper Laboratory Technical Report*, No. R-1258.

[2] **Acton, Forman S. ,**
"Numerical Methods That Work," Harper and Row, Publishers, New York, 1970.

[3] **Alberts, T.E., Hastings, G. G., Book, W.J., and Dickerson, S.L.,**
"Experiments in Optimal Control of a Flexible Arm with Passive Damping," *Fifth VPISSU/AIAA Symposium on Dynamics and Control of Large Structures*, June 12, 1985, Blacksburg, VA, pp 423-435.

[4] **Alfriend, K. T.; and Longman, R. W.,**
"Rotational Maneuvers of Large Flexible Spacecraft," *Fix this ref ****** Guidance and Control*, 1980, Vol. 42,

[5] **An, Chae H, et al.,**
"Experimental Evaluation of Feedforward and Computed Torque Control," *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, March 31–April 3, 1987, Raleigh, North Carolina, pp 165-168.

[6] **Asada, Haruhiko; Ma, Zeng-Dong; and Tokumaru, Hidekatsu,**
"Inverse Dynamics of Flexible Robot Arms for Trajectory Control.," *Modeling and Control of Robotic Manipulators.* , 1987 ASME Winter Annual Meeting. pp 329–336.

[7] **Aspinwall, D. M.,**
"Acceleration Profiles for Minimizing Residual Response," *Journal of Dynamic Systems, Measurement, and Control*, March 1980, Vol. 102, No. 1, pp 3–6.

[8] **Bailey, Thomas; and Hubbard, James, E.,**
"Distributed Piezoelectric-Polymer Active Vibration Control of a Cantilever Beam," *Journal of Guidance and Control*, Vol. 8, No. 5, Sept-Oct. 1985, pp 605-611.

[9] **Balas, Mark J.,**
"Active Control of Flexible Systems," *Proceedings of the 1977 Symposium on Dynamics and Control of Large Flexible Spacecraft*, June 13-15, 1977, Blacksburg, VA, pp 217–236.

[10] **Balas, Mark J.,**
"Feedback Control of Flexible Systems," *IEEE Transactions on Automatic Controls*, Vol. 23, 1978, pp 673–679.

[11] **Balas, Mark J.; and Canavin, Joseph R.,**
"An active Modal Control System Philisophy for a Class of Large Space Structures," *Proceedings of the 1977 Symposium on Dynamics and Control of Large Flexible Spacecraft*, June 13-15, 1977, Blacksburg, VA, pp 271–285.

[12] **Balestrino, A., De Maria, G., and Sciavicco, L.,**
"An Adaptive Model Following Control for Robotic Manipulators," *Journal of Dynamic Systems, Measurement, and Control*, September, 1983, Vol. 105, pp 143–51.

[13] **Barraco, A., Cuny, B., Ishiomin, G.,**
"Dynamic Models for Flexible Robots: Different Approaches," *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, April 7–10, 1986, San Francisco, CA, pp 1038–1043.

[14] **Bathe, Klaus-Jürgen,**
"Finite Element Procedures in Engineering Analysis," Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982, pp 102–113.

[15] **Bayo, E.,**
"Computed Torque for the Position Control of Open-Chain Flexible Robots," *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, April 25–29, 1988, Philadelphia, Pennsilvania, pp 316–321.

[16] **Bayo, E.; and Paden,B.,**
"On Trajectory Generation for Flexible Robots," *Journal of Robotic Systems*, February, 1987, Vol. 4, No. 2, pp 229–35.

[17] **Biswas, S. K., and Klafter, R. D.,**
"Dynamic Modeling and Optimal Control of Flexible Robotic Manipulators," *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, April 25–29, 1988, Philadelphia, Pennsilvania, pp 15-20.

[18] **Blevins, Robert D.,**
"Formulas for Natural Frequency and Mode Shape," Van Nostrand Reinhold Company, New York, 1979.

[19] **Bolz, Ray E., and Tuve, George L.,**
"CRC Handbook of Tables for Applied Engineering Science," CRC Press, Inc., Boca Raton, Florida, 1973, pp 1071.

[20] **Book, W.J.,**
"Analysis of Massless Elastic Chains with Servo Controlled Joints," *Journal of Dynamic Systems, Measurement, and Control*, September, 1979, Vol. 101, pp 187-192.

[21] **Book, Wayne J.; and Cetinkunt, Sabri,**
"Near Optimal Control of Flexible Robot Arms on Fixed Paths," *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, April 1985, St. Louis, MO, pp 1522–28.

[22] **Book, W. J.; Maizza-Neto, O.; and Whitney, D. E. ,**
"Feedback Control of Two Beam, Two Joint Systems with Distributed Flexibility," ASME 75-WA/Aut-26.

[23] **Book, W. J.; Majette, M.,**
"A Comparison of Natural Frequency Prediction Methods for Flexible Manipulator Arms," ASME 80-WA/DSC-19.

[24] **Breakwell, J.A.,**
"Optimal Feedback Slewing of Flexible Spacecraft," *Journal of Guidance and Control*, September-October 1981, Vol. 4, No. 5, pp 472-479.

[25] **Bremer, H.,**
"On the Dynamics of Flexible Manipulators," *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, March 31–April 3, 1987, Raleigh, North Carolina, pp 1556-1560.

[26] **Burns, John A.; and Cliff, Eugene M.,**
"On the Formulation of some Distributed System Parameter Identification Problems," *Proceedings of the 1977 Symposium on Dynamics and Control of Large Flexible Spacecraft*, June 13-15, 1977, Blacksburg, VA, pp 87–105.

[27] **Cannon, Robert H. and Schmitz, Eric.,**
"Initial Experiments on the End-Point Control of a Flexible One Link Robot," *The International Journal of Robotics Research*, Fall 1984, Vol. 3, No. 3.

[28] **Cannon, Robert H. and Rosenthal, Dan E.,**
"Experiments in Control of Flexible Structures with Noncolocated Sensors and Actuators," *Journal of Guidance and Control*, Vol. 7, No. 5, pp 546–553.

[29] **Cannon, Robert H. and Schmitz, Eric.,**
"Precise Control of Flexible Manipulators," *Journal of Robotics*, Pre-print March 1983.

[30] **Cetinkunt, Sabri, and Book, Wayne,**
"Symbolic Modeling of Flexible Manipulators," *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, March 31–April 3, 1987, Raleigh, North Carolina, pp 2074-2080.

[31] **Chalhoub, N. G., and Ulsoy, A. G.,**
"Control of a Flexible Robot Arm: Experimental and Theoretical Results," *Journal of Dynamic Systems, Measurement, and Control*, December 1987, Vol. 109, pp 299–309.

[32] **Christian, Andrew D.; and Seering, Warren P.,**
"Design Considerations for an Earth Based Flexbile Robotic System," *Submitted to the 1989 IEEE International Conference on Robotics and Automation*, May 14 – 19 Scottsdale, Arizona

[33] **Chun, Hon M.; Turner, James D.; and Juang, Jer-Nan,**
"Disturbance-Accommodating Tracking Maneuvers of Flexible Spacecraft," *Journal of the Astronautical Sciences*, April-June 1985, Vol. 33, No. 2, pp 197-216.

[34] **Crandall, Stephen H.; Karnopp, Dean C.; Kurtz, Edward F., Jr.; and Pridmore-Brown, David C.,**
"Dynamics of Mechanical and Electromechanical Systems," Robert E. Krieger Publishing Company, Malabar, Florida, 1982, pp 332–351.

[35] **Crawley, Edward F., de Luis, J.,**
"Use of Piezo-Ceramics as Distributed Actuators in Large Space Structures," *Structures, Structural Dynamics and Materials Conference*, April 15–17, 1986, Orlando, FL, *AIAA Paper No. 85-0626*, pp 126-133.

[36] **Crawley, Edward F., de Luis, J.,**
"Experimental Verification of Distributed Piezoelectric Actuators for use in Precision Space Structures," *Structures, Structural Dynamics and Materials Con-*

ference, May 19–21, 1986, San Antonio, TX, *AIAA Paper No. 86-0878*, pp 116-124.

[37] **Dado, Mohammad H.F., and Soni, A.H.,**
"Dynamic Response Analysis of 2-R Robots with Flexible Joints," *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, March 31–April 3, 1987, Raleigh, North Carolina, pp 479-483.

[38] **Daniel, R. W.; Irving, M. A.; Fraser, A. R.; and Lambert, M.,**
"The control of Compliant Manipulator Arms," *??*, pp 225–231.

[39] **De Maria, Giuseppe, and Siciliano, Bruno,**
"A Multilayer Approach to Control of a Flexible Arm," *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, March 31–April 3, 1987, Raleigh, North Carolina, pp 774-778.

[40] **Dubowsky, Steven,**
"Active Control of Mechanical Systems: The State-of-the-Art for Robotic Manipulators," , 1985, pp 258-261.

[41] **Dubowsky, Steven, and DesForges, D. T.,**
"The Application of Model–Referenced Adaptive Control to Robotic Manipulators," *Journal of Dynamic Systems, Measurement, and Control*, September, 1979, Vol. 101, pp 193–200.

[42] **Dubowsky, S.; and Shiller, Z.,**
"Optimal Dynamic Trajectories for Robotic Manipulators," *Theory and Practice of Robots and Manipulators. Proceedings of RoManSy 1984, The Fifth CISM–IFToMM Symposium*, MIT Press, Cambridge, MA,

[43] **Eldred, Daniel, and Schaechter, David,**
"Experimental Demonstration of Static Shape Control," *Journal of Guidance and Control*, Vol. 6, No. 3, May-June 1983, pp 188–92.

[44] **Eppinger, Steven D. ,**
"," Massachusetts Institute of Technology, PhD Thesis,

[45] **Farrenkopf, R.L.,**
"Optimal Open-Loop Maneuver Profiles for Flexible Spacecraft," *Journal of Guidance and Control*, Vol. 2, No. 6, November–December 1979, pp 491–498.

[46] **Forward, Robert L.; Swigert, Charles J.; and Obal, Michael.,**
"Electronic Damping of a Large Optical Bench," *Shock and Vibration Bulletin*, No. 4, May 1983, pp 51-61.

[47] **Gebler, Bernd,**
"Feed-forward Control Strategy for an Industrial Robot with Elastic Links and Joints," *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, March 31–April 3, 1987, Raleigh, North Carolina, pp 923-928.

[48] **Gieck, Kurt,**
"Engineering Formulas.," McGraw-Hill Book Company, Inc., New York, 1983, pp E4.

[49] **Gupta, Narendra K.,**
"Frequency-Shaped Cost Functionals: Extension of Linear-Quadratic-Gaussian Design Methods," *Journal of Guidance and Control*, Vol. 3, No. 6, Nov.-Dec 1980, pp 529–35.

[50] **Hale, A. L.; and Meirovitch, L.,**
"A Special Purpose Symbolic Manipulation Program for the Derivation for the Equations of Motion for Large Flexible Structures," *Proceedings of the 1977 Symposium on Dynamics and Control of Large Flexible Spacecraft*, June 13-15, 1977, Blacksburg, VA, pp 367–382.

[51] **Hanagud, S.; Obal, M. W.; and Meyyappa, M.,**
"Electronic Damping Techniques and Active Vibration Control," *Structures, Structural Dynamics and Materials Conference*, 1985, pp 443-450.

[52] **Hastings, Gordon G., and Book, Wayne J.,**
"Experiments in Optimal Control of a Flexible Arm," *Proceedings of the 1985 American Control Conference*, 1985, pp 728-729.

[53] **Hastings, Gordon G.; Dorsey, John F.; and Book, Wayne J.,**
"Application of Balanced Realizations to Estimate Model Order Requirements for Flexible Manipulators," *ASME Winter Annual Meeting*, 1987, Boston, MA,

[54] **Hattis, P.; Lepanto, J.; Sargent, D.; Turnbull, J.,**
"A Survey of the Payload Deployment and Retrieval System Environment," *Charles Stark Draper Laboratory Technical Report*, No. CSDL-P-1838, January 1984.

[55] **Hollars, Michael G. and Cannon, Robert H.,**
"Initial Experiments on the End-Point Control of a Two Link Manipulator with Flexible Tendons," *ASME Winter Annual Meeting*, November 19, 1985.

[56] **Hollars, Michael G. and Cannon, Robert H.,**
"Experiments on the End-Point Control of a Two-Link Robot with Elastic Drives," *AIAA Paper 86-1977*, 1986.

[57] **Hollis, Ralph L.,**
"A Planar XY Robotic Fine Positioning Device," *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, April 1985, St. Louis, MO.

[58] **Hudson, J. E.,**
"Adaptive Array Principles," Institute of Electrical Engineers, New York 1981.

[59] **Jones, James F., Petterson, Ben J.,**
"Oscillation Damped Movement of Suspended Objects," *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, April 25–29, 1988, Philadelphia, Pennsilvania, pp 956–962.

[60] **Juang, Jer-Nan,**
"Optimal Design of a Passive Vibration Absorber for a Truss Beam," *AIAA Journal of Guidance and Control*, Vol. 7, No. 6, November-December 1984, pp 733-739.

[61] **Juang, Jer-Nan; Turner, James D.; and Chun, Hon M.,**
"Closed-Form Solutions for Feedback Control with Terminal Constraints," *Journal of Guidance and Control*, January–February 1985, Vol. 8, No. 1, pp 39–43.

[62] **Judd, Robert P., and Faulkenburg, Donald R.,**
"Dynamics of Nonrigid Articulated Robot Linkages," *IEEE Transactions on Automatic Control*, Vol. AC-30, No. 5, May 1985, pp 499-502.

[63] **Junkins, John L.; Turner, James D.,**
"Optimal Spacecraft Rotational Maneuvers," Elsevier Science Publishers, New York, 1986.

[64] **Kane, Thomas R.; and Levinson, David A.,**
"Dynamics: Theory and Applications," McGraw-Hill, Inc., New York, 1985.

[65] **Kane, Thomas R.; Likins, Peter W.; Levinson, David A.,**
"Spacecraft Dynamics," McGraw-Hill, Inc., New York, 1983.

[66] **Kane, T.R., Ryan, R.R., Banerjee, A.K.,**
"Dynamics of a Cantilever Beam Attached to a Moving Base," *Journal of Guidance, Control, and Dynamics*, March–April 1987, Vol. 10, No. 2, pp 139–151.

[67] **Kanoh, Hideaki, and Lee, Ho Gil,**
"Vibration Control of One-Link Flexible Arm," *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, April 1985, St. Louis, MO, pp 1172–77.

[68] **Karkkainen, Paavo,**
"Compensation Manipulator Flexibility Effects by Modal Space Techniques," *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, April 1985, St. Louis, MO, pp 972–77.

[69] **Khorasani, K., and Spong, Mark W.,**
"Invariant Manifolds and Their Application to Robot Manipulators with Flexible Joints," *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, April 1985, St. Louis, MO, pp 978–83.

[70] **Korolov, V. V., Chen, Y. H.,**
"Robust Control of a Flexible Manipulator Arm," *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, April 25–29, 1988, Philadelphia, Pennsilvania, pp 159-161.

[71] **Kosut, Robert L.; Salzwedel, Horst; and Emami-Naeini, Abbas,**
"Robust Control of Flexible Spacecraft," *Journal of Guidance, Control, and Dynamics*, March-April, 1983, Vol. 6, No. 2, pp 104-111.

[72] **Kotnik, P. T.; Yurkovich, S.; and Ozguner, U., ,**
"Acceleration Feedback for control of a flexible Manipulator Arm," *Journal of Robotic Systems*, Vol. 5, No. 3, June 1988, pp (to appear).

[73] **Krishnan, H., and Vidyasagar, M.,**
"Control of a Single-Link Flexible Beam Using a Hankel-Norm-Based Reduced Order Model," *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, April 25–29, 1988, Philadelphia, Pennsylvania, pp 9-14.

[74] **Lanczos, Cornelius,**
"The Variational Principles of Mechanics," University of Toronto Press, Toronto, 1949, pp 35–72.

[75] **Larson, V.; and Likins, P.,**
"An Application of Modern Control Theory to an Elastic Spacecraft," *Proceeding of the Symposium on Dynamics and Control of Non-Rigid Space Vehicles*, Frascati, Italy, May 1976, pp 221-226.

[76] **Laurenson, Robert M.; and Heaton, Paul W.,**
"Equations of Motion for a Rotating Flexible Structure," *Proceedings of the 1977 Symposium on Dynamics and Control of Large Flexible Spacecraft*, June 13-15, 1977, Blacksburg, VA, pp 341–350.

[77] **Lim, Jae, editor,**
"Speech Enhancement," Prentice-Hall, Inc., Englewood Cliffs, NJ, 1983.

[78] **Liu, Chang P.; and Forward, Robert L.,**
"Electronic Damping of Resonance in Gimbal Structures," *AIAA Paper*, No. 81-0556, pp 219-226.

[79] **Low, K. H.; and Vidyasagar, M.,**
"A Lagrangian Formulation of the Dynamic Model for Flexible Manipulator Systems," *Journal of Dynamic Systems, Measurement, and Control*, June, 1988, Vol. 110, pp 175–181.

[80] **Lozano-Perez, Tomas,**
"A Simple Motion-Planning Algorithm for General Robot Manipulators," *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 3 June 1987, pp 224–38.

[81] **Marino, R., and Spong, M. W.,**
"Nonlinear Control Techniques for Flexible Joint Manipulators: A Single Link," *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, April 7–10, 1986, San Francisco, CA, pp 1030–1036.

[82] **Markley, F. L.,**
"Large Angle Maneuver Strategies for Flexible Spacecraft," *AAS/AIAA Astrodynamics Conference*, Provencetown, MA, June 1979, pp 625–647.

[83] **Martin, J.; Ozguner, U.; and Yurkovich, S.,**
"An Active Vibration–Damper for Flexible Structures," *Proceedings of the Seventh Annual Conference on Modeling and Simulation*, April 1986, Pittsburgh, PA, pp 687–92.

[84] **Meckl, P. and Seering, W.,**
"Minimizing Residual Vibration for Point-to-point Motion," *ASME Journal of Vibration, Acoustics, Stress, and Reliability in Design*, Vol. 107, No. 4, October, 1985, pp 378-382.

[85] **Meckl, P. and Seering, W.,**
"Active Damping in a Three-Axis Robotic Manipulator," *Journal of Vibration, Acoustics, Stress, and Reliability in Design*, Vol. 107, No. 1, January 1985, pp 38-46.

[86] **Meckl, Peter H., and Seering, Warren P.,**
"Feedforward Control Techniques To Achieve Fast Settling Time in Robots," *Proceedings of The American Controls Conference*, Seatle, Washington, June, 1986.

[87] **Meckl, Peter H., and Seering, Warren P.,**
"Reducing Residual Vibration in Systems with Time Varying Resonances," *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, March 31–April 3, 1987, Raleigh, North Carolina, pp 1690-1695.

[88] **Meckl, Peter H., and Seering, Warren P.,**
"Controlling Velocity–Limited Systems to Reduce Residual Vibration," *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, April 25–29, 1988, Philadelphia, Pennsilvania.

[89] **Meckl, Peter H.,**
"Control of Vibration in Mechanical Systems Using Shaped Reference Inputs," *PhD Thesis, Massachusetts Institute of Technology*, June 1988. also **AI-TR-1018**, The Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts.

[90] **Meirovitch, L.; Van Landingham, H. F.; and Oz, H.,**
"Distributed Control of Spinning Flexible Spacecraft," *Proceedings of the 1977 Symposium on Dynamics and Control of Large Flexible Spacecraft*, June 13-15, 1977, Blacksburg, VA, pp 249–269.

[91] **Miller, D.W., Crawley, E.F., and Ward, B.A.,**
"Inertial Actuator Design for Maximum Passive and Active Energy Dissipation in Flexible Space Structures," *Structures, Structural Dynamics and Materials Conference*, April 15–17, 1986, Orlando, FL, *AIAA Paper No. 85-0777*, pp 536-544.

[92] **Monzingo, Robert A.; Miller, Thomas W.,**
"Introduction to Adaptive Arrays," John Wiley and Sons, New York 1980.

[93] **Mulgrew, Bernard; Cowan, Colin F. N.,**
"Adaptive Filters and Equalizers," Kluwer Academic Publishers, Boston 1988.

[94] **Longman, R. W.; and Alfriend, K. T.,**
"Optimal Control of Large Angle Attitude Maneuvers for Flexible Spacecraft ," *Proceedings of the 8th IFAC Symposium*, Oxford, England, July 1979, pp 57–62.

[95] **Naganathan, G., and Soni, A.H.,**
"An analytical and Experimental Investigation of Flexible Manipulator Performance," *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, March 31–April 3, 1987, Raleigh, North Carolina, pp 767-773.

[96] **Naganathan, G., and Soni, A.H.,**
"Non-linear Flexibility Studies for Spatial Manipulators," *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, April 7–10, 1986, San Francisco, CA, pp 373-378.

[97] **Narasimhan, S., Siegel, D. M., Hollerbach, J. M.,**
" A Standard Architecture for Controlling Robots," **AIM No. 977**, The Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts. June 1988.

[98] **Narasimhan, S., Siegel, D. M., Hollerbach, J. M.,**
" Condor: A Revised Architecture for Controlling the Utah-MIT Hand," *Proc. IEEE Conference on Robotics and Automation*, Vol. 1, pp 446-450. April 1988.

[99] **Nelson, Winston L.; Mitra, Debasis; and Boie, Robert A.,**
"End-Point Sensing and Load-Adaptive Control of a flexible Robot Arm," , Ft. Lauderdale, FL, 1985, pp 1410-1415.

[100] **Nesline,F. William; Zarchan, Paul,**
"Why Modern Controllers Can Go Unstable in Practice," *Journal of Guidance and Control*, July-August 1984, Vol. 7, No. 4, pp 495-500.

[101] **Newland, D. E. ,**
"An Introduction to Random Vibrations and Spectral Analysis," Longman Group Limited, New York, 1984, pp 146–147.

[102] **Nicosia, S., Tomei, P., and Tornambe, A.,**
"Dynamic Modeling of Flexible Robot Manipulators," *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, April 7–10, 1986, San Francisco, CA, pp 365-372.

[103] **Nurre, G.S.; Ryan, R.S.; Scofield, H.N.; and Sims, J.L.,**
"Dynamics and Control of Large Space Structures," *Journal of Guidance and Control*, September-October 1984, Vol. 7, No. 5, pp 514-526.

[104] **Ogata, Katsuhiko,**
"Modern Control Engineering," Prentice-Hall, Inc., Englewood Cliffs, NJ, 1970, pp 282.

[105] **Oppenheim, Alan V., and Schafer, Ronald W.,**
"Digital Signal Processing," Prentice-Hall, Inc., Englewood Cliffs, NJ, 1975.

[106] **Ormondroyd, J., Den Hartog, J.P.,**
"The Theory of the Dynamic Vibration Absorber," *Transactions of the ASME, APM-50-7*, 1928.

[107] **Parks, T. W., McClellan, J. H.,**
"Chebyshev Approximation for Nonrecursive Digital Filters with Linear Phase," *IEEE Transactions on Circuit Theory,* Vol. CT-19, March 1972, pp 189–94.

[108] **Paul, Richard P.,**
"Robot Manipulators," MIT Press, Cambridge, MA, 1982, pp 139–43.

[109] **Pfeiffer, F., and Gebler, B.,**
"A Multistage-Approach to the Dynamics and Control of Elastic Robots," *Proceedings of the 1988 IEEE International Conference on Robotics and Automation,* April 25–29, 1988, Philadelphia, Pennsilvania, pp 2-8.

[110] **Plump, J. M.; Hubbard, J. E.; and Bailey, T.,**
"Nonlinear Control of a Distrubuted System: Simulation and Experimental Results," *Journal of Dynamic Systems, Measurement, and Control,* June, 1987, Vol. 109, pp 133-139.

[111] **Poelaert, Daniel H. L. ,**
"Dynamic Analysis of a Non-Rigid Spacecraft," *Proceedings of the 1977 Symposium on Dynamics and Control of Large Flexible Spacecraft,* June 13-15, 1977, Blacksburg, VA, pp 351–366.

[112] **Prucz, Z.; Soong, T.T.; and Reinhorn, A.,**
"An Analysis of Pulse Control for Simple Mechanical Systems," *Journal of Dynamic Systems, Measurement, and Control,* June, 1985, Vol. 107, pp 123–131.

[113] **Rabiner, Lawrence R., McClellan, James H., Parks, Thomas, W.,**
"FIR Digital Filter Design Techniques Using Weighted Chebyshev Approximation," *Proceeding of the IEEE,* Vol. 63, No. 4, April, 1975, pp 595–610.

[114] **Rakhsha, F., and Goldenberg, A. A.,**
"Dynamic Modeling of a Single-Link Flexible Robot," *Proceedings of the 1985 IEEE International Conference on Robotics and Automation,* April 1985, St. Louis, MO, pp 984–989.

[115] **Sakawa, Yoshiyuki; Matsuno, Fumitoshi,; and Fukushima, Shigenobu,**
"Modeling and Feedback Control of a Flexible Arm," *Journal of Robotic Systems,* Vol. 2, No. 4, pp 453–472.

[116] **Schaechter, David,**
"Hardware Demonstration of Flexible Beam Control," *Journal of Guidance and Control,* Vol. 5, No. 1, Jan-Feb 1982, pp 48–53.

[117] **Sehitoglu, H.; and Aristizabal, J. H.,**
"Design of a Trajectory Controller for Industrial Robots Using Bang-Bang and Cycloidal Motion Profiles," *Robotics: Theory and Applications*, ASME Winter Annual Meeting, Anaheim, CA, December 1986, pp 169–175.

[118] **Seraji, H.,**
"An adaptive Cartesian Control Scheme for Manipulators," *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, March 31–April 3, 1987, Raleigh, North Carolina, pp 157-164.

[119] **Seraji, H.,**
"A New Approach to Adaptive Control of Manipulators," *Journal of Dynamic Systems, Measurement, and Control*, September, 1987, Vol. 109, pp 193–202.

[120] **Shung, I.Y., and Vidyasagar, M.,**
"Control of a Flexible Robot Arm with Bounded Input: Optimum Step responses," *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, March 31–April 3, 1987, Raleigh, North Carolina, pp 916-922.

[121] **Sidman, Michael David,**
"Adaptive Control of a Flexible Structure," *Stanford University PhD Thesis*, 1986.

[122] **Singer, Neil C.; Seering, Warren P. ,**
"Controlling Vibration in Remote Manipulators," *ASME Design Automation Conference*, Boston, MA, September 27–30, 1987, Vol. 2, pp 11.

[123] **Singer, Neil C.; Seering, Warren P. ,**
"Using Acausal Shaping Techniques to Reduce Robot Vibration," *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, Philadelphia, PA, April 25–29, 1988, pp 1434–7.

[124] **Singer, Neil C.; Seering, Warren P. ,**
"Preshaping Command Inputs to Reduce System Vibration," *Submitted to the ASME Journal of Dynamic Systems Measurement and Control*, March, 1988. also **AIM No. 1027**, The Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts.

[125] **Singh, Sahjendra N., and Schy, Albert A.,**
"Decomposition and State Variable Feedback Control of Elastic Robotic Systems," *Proceedings of the 1985 American Control Conference*, 1985, Boston, MA, pp 375–80.

[126] **Slotine, J.-J. E.; Hedrick, J. K.; Misawa, E. A.,**
"On Sliding Observers for Nonlinear Systems," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 109, September 1987, pp 245–51.

[127] **Smith, O.J.M.,**
"Feedback Control Systems," McGraw-Hill Book Company, Inc., New York, 1958.

[128] **Stearns, Samuel D.,**
"Digital Signal Analysis," Hayden Book Co., Rochelle Park, New Jersey, 1975.

[129] **Sunada, W.H.; and Dubowsky, S. ,**
"On the Dynamic Analysis and Behavior of Industrial Robotic Manipulators with Elastic Members," *Transactions of the ASME*, Vol. 105, March 1983.

[130] **Swigert, C.J.,**
"Shaped Torque Techniques," *Journal of Guidance and Control*, September-October 1980, Vol. 3, No. 5, pp 460-467.

[131] **Swigert, Charles J.; and Forward, Robert L.,**
"Electronic Damping of Orthogonal Bending Modes in a Cylindrical Mast – Theory," *Journal of Spacecraft*, Vol. 18, No. 1, January–February, 1981, pp 5-10.

[132] **Truckenbrodt, A. Von,**
"Regelung elastischer mechanischer Systeme," *Regelungstechnik*, Vol. 30, No. 8, August 1982, pp 277–85.

[133] **Turnbull, Joseph F. ,**
"Acceleration Environment of Payloads While Being Handled by the Shuttle Remote Manipulator System," *Charles Stark Draper Laboratory Technical Report* , No. CSDL-P-1732.

[134] **Tzes, A., and Yurkovich, S.,**
"A Sensitivity Analysis Approach to Control of Manipulators with Unknown Load," *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, March 31–April 3, 1987, Raleigh, North Carolina, pp 496–502.

[135] **Usoro, P. B., Nadira, R., and Majil, S. S.,**
"Control of Lightweight Flexible Manipulators: A Feasibility Study," *Proceedings of the 1984 American Control Conference*, San Diego, CA, pp 1209–16.

[136] **Wang, David, and Vidyasagar, M.,**
"Control of a Flexible Beam for Optimum Step Response," *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, March 31–April 3, 1987, Raleigh, North Carolina, pp 1567-1572.

[137] **Wang, S.; Hsia, T. C.; Wiederrich, J. L.,**
"Open-Loop Control of a Flexible Robot Manipulator," *International Journal of Robotics and Automation*, Vol. 1, No. 2, 1986, pp 54–57.

[138] **Widmann, Glenn R., and Ahmad, Shaheen.,**
"Control of Industrial Robots with Flexible Joints," *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, March 31–April 3, 1987, Raleigh, North Carolina, pp 1561-1566.

[139] **Yamada, I.; and Nakagawa, M.,**
"Reduction of Residual Vibrations in Positioning Control Mechanism," *Journal of Vibration, Acoustics, Stress, and Reliability in Design*,

[140] **Yang, Guo-Ben; and Donath, Max,**
"Dynamic Model of a One-Link Robot Manipulator with both Structural and Joint Flexibility," *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, April 25–29, 1988, Philadelphia, Pennsilvania, pp 476–81.

[141] **Yurkovich, S.; Ozguner, U.; and Al–Abbass, F.,**
"Model Reference, Sliding Mode Adaptive Control for Flexible Structures," *Journal of the Astronautical Sciences*, Vol. 36, No. 3, July–September, 1988, pp (to appear).

[142] **Yurkovich, S.; Ozguner, U.; Tzes, A.; and Kotnik, P. T.,**
"Flexible Manipulator Control Experiments and Analysis," *Proceedings of the Workshop on Space Telerobotics*, Vol. 111, NASA–JPL (JPL Pub 87–13), July 1, 1987, pp 279–287.

CS-TR Scanning Project
Document Control Form

Date : 9 / 14 / 95

Report # AI-TR-1030

Each of the following should be identified by a checkmark:
Originating Department:

☒ Artificial Intellegence Laboratory (AI)
☐ Laboratory for Computer Science (LCS)

Document Type:

☒ Technical Report (TR) ☐ Technical Memo (TM)
☐ Other: ____________________

Document Information

Number of pages: 241 (248-IMAGES)
Not to include DOD forms, printer intstructions, etc... original pages only.

Originals are:
☒ Single-sided or
☐ Double-sided

Intended to be printed as :
☐ Single-sided or
☒ Double-sided

Print type:
☐ Typewriter ☐ Offset Press ☒ Laser Print
☐ InkJet Printer ☐ Unknown ☐ Other: ____________

Check each if included with document:

☒ DOD Form (2) ☐ Funding Agent Form ☒ Cover Page
☐ Spine ☐ Printers Notes ☐ Photo negatives
☐ Other: ____________________

Page Data:

Blank Pages(by page number): FOLLOW TITLE PAGE, i i, vii ,

Photographs/Tonal Material (by page number): ____________

Other (note description/page number):

Description : Page Number:

(A) IMAGE MAP: (1-14) UN#'ED TITLE & BLANK, ii-iv, UN#'ED BLANK,
V-Vii, UN#'ED BLANK, Viii-X, UN#'ED LIST OF FIGS.
(15-241) PAGES #'ED 1-227
(242-248) SCANCONTROL, COVER, DOD (2), TRGTS (3)
(B) CUT & PASTE FIG'S ON PAGES #'ED iii-iv.

Scanning Agent Signoff:

Date Received: 9 / 14 / 95 Date Scanned: 9 / 15 / 95 Date Returned: 9 / 21 / 95

Scanning Agent Signature: Michael W. Cook

Rev 9/94 DS/LCS Document Control Form cstrform.vsd

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AI-TR 1030	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Residual Vibration Reduction in Computer Controlled Machines		5. TYPE OF REPORT & PERIOD COVERED technical report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Neil Singer		8. CONTRACT OR GRANT NUMBER(s) N00014-86-K-0685 N00014-85-K-0124
9. PERFORMING ORGANIZATION NAME AND ADDRESS Artificial Intelligence Laboratory 545 Technology Square Cambridge, MA 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency 1400 Wilson Blvd. Arlington, VA 22209		12. REPORT DATE February 1989 13. NUMBER OF PAGES 238
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Office of Naval Research Information Systems Arlington, VA 22217		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

vibrations
oscillations
teleoperators
flexible manipulators

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Control of machines that exhibit flexibility becomes important when designers attempt to push the state of the art with faster, lighter machines. Three steps are necessary for the control of a flexible plant. First, a good model of the plant must exist. Second, a good controller must be designed. Third, inputs to the controller must be constructed using knowledge of the system dynamic response. There is a great deal of literature pertaining to modeling and control but little dealing with the shaping of system inputs. See back

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block 20 continued.

Chapter 2 examines two input shaping techniques based on frequency domain analysis. The first involves the use of the first derivative of a gaussian exponential as a driving function template. The second, acausal filtering, involves removal of energy from the driving functions at the resonant frequencies of the system. Chapter 3 presents a linear programming technique for generating vibration-reducing driving functions for systems. Chapter 4 extends the results of the previous chapter by developing a direct solution to the new class of driving functions. A detailed analysis of the new technique is presented from five different perspectives and several extensions are presented. Chapter 5 verifies the theories of the previous two chapters with hardware experiments. Because the new technique resembles common signal filtering, chapter 6 compares the new approach to eleven standard filters. The new technique will be shown to result in less residual vibration, have better robustness to system parameter uncertainty, and require less computation than other currently used input shaping techniques.